

ADAU34232



**FINAL REPORT  
AUTOMATIC THRESHOLD DETECTOR TECHNIQUES**

N. Freedman, P. Cornwell, and P. Barr  
Raytheon Company  
Equipment Development Laboratories  
Advanced Development Laboratory  
Wayland, Massachusetts 01778

15 July 1976



APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.



**U.S. ARMY MISSILE COMMAND**

**Redstone Arsenal, Alabama 35802**

*Prepared for:*

Advanced Sensors Directorate  
US Army Missile Research, Development  
and Engineering Laboratory  
US Army Missile Command  
Redstone Arsenal, Alabama 35809

#### **DISPOSITION INSTRUCTIONS**

**DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.**

#### **DISCLAIMER**

**THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.**

#### **TRADE NAMES**

**USE OF TRADE NAMES OR MANUFACTURERS IN THIS REPORT DOES NOT CONSTITUTE AN OFFICIAL INDORSEMENT OR APPROVAL OF THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Automatic Threshold Detector Techniques.		5. TYPE OF REPORT & PERIOD COVERED Final rept.
6. AUTHOR(s) Freedman, P./Cornwell, P./Barr Raytheon Company Donald W./Burlage (COTR) U.S. Army Missile Command		7. PERFORMING ORG. REPORT NUMBER ER76-4288
8. PERFORMING ORGANIZATION NAME AND ADDRESS Raytheon Company Equipment Development Laboratories Advanced Development Laboratory Wayland, Massachusetts 01778		9. CONTRACT OR GRANT NUMBER(s) DAAH01-76-C-0363
10. CONTROLLING OFFICE NAME AND ADDRESS Commander, US Army Missile Command Attn: DRSMI-RPR Redstone Arsenal, AL 35809		11. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1M362303 A214 AMCHS 632303.11-21401
12. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 123p.		13. REPORT DATE 15 July 1976
14. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		15. NUMBER OF PAGES 119
15. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		16. SECURITY CLASS. (of this report) UNCLASSIFIED
16. SUPPLEMENTARY NOTES		17. DECLASSIFICATION/DOWNGRADING SCHEDULE
18. KEY WORDS (Continue on reverse side if necessary and identify by block number) Radar False Alarms Constant False Alarm Rate Signal Processing Threshold Control		
19. ABSTRACT (Continue on reverse side if necessary and identify by block number) A study was performed to develop and evaluate techniques for suppressing false alarms in air-defense radars. The study included analytical performance investigations for a number of candidate signal processor architectures and cost/performance tradeoffs for the leading candidates. Detail analyses of processor performance were conducted as a function of parameters such as number of target samples coherently processed, input signal-to-clutter ratio, input signal-to-noise ratio, and presence or absence of Moving-Target-Indicator canceller processing. Detailed block diagrams were generated and estimates made.		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

298 353 AB

**UNCLASSIFIED**

**SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)**

of such cost-influencing factors as the number of components/cards required by each approach and software requirements, as a function of system specifications such as number of instrumented range cells and number of samples coherently processed.

NVS		W/LS Section	<input checked="" type="checkbox"/>
C		B/L Section	<input type="checkbox"/>
Date: 10/27			
BY: [Signature]			
[Signature]			

**UNCLASSIFIED**

**SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)**

**FINAL REPORT**

**AUTOMATIC THRESHOLD DETECTOR TECHNIQUES**

**Contract No. DAAH01-76-C-0363**

**ER76-4208**

**15 July 1976**

**Prepared for:**

**HEADQUARTERS  
U.S. Army Missile Command  
Redstone Arsenal, Alabama 35809**

**Prepared by:**

**RAYTHEON COMPANY  
Equipment Development Laboratories  
Advanced Development Laboratory  
Wayland, Massachusetts 01778**

## TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
1. INTRODUCTION	1-1
1.1 Purpose of the Study	1-1
1.2 Organization of the Study	1-1
1.2.1 Background	1-1
1.2.2 General Processor Configuration	1-1
1.2.3 Analysis	1-2
1.2.4 Hardware/Software Configurations and Tradeoffs	1-2
2. BACKGROUND AND ASSUMPTIONS	2-1
2.1 The Experimental Array Radar (EAR)	2-1
2.2 Limits on the Processor Design	2-2
2.2.1 Scan Time Limitation	2-2
2.2.2 Range Walk	2-3
2.2.3 Target Acceleration	2-3
2.2.4 Target Coherence	2-4
2.3 The Clutter Environment	2-6
2.3.1 Ground Clutter	2-6
2.3.2 Precipitation Clutter	2-6
3. THE GENERAL APPROACH	3-1
3.1 MTI	3-1
3.2 Input Weighting	3-3
3.3 Input Buffer	3-3
3.4 The Fast-Fourier Transform (FFT) Processor	3-3
3.5 Magnitude Determination	3-3
3.6 Range-Cell CFAR	3-4
3.7 Frequency-Bin CFAR	3-4
3.8 Clutter Map	3-4
3.9 Threshold	3-4
4. ANALYTICAL FTS STUDY	4-1
4.1 Introduction	4-1
4.2 Theoretical Model	4-3
4.2.1 Signal Models	4-3
4.2.2 Clutter Map/CFAR Concept	4-6
4.2.3 Detection Probability Model	4-8
4.2.4 False Alarm Probabilities	4-12
4.3 Clutter Map/CFAR Computer Simulation	4-16
4.3.1 Introduction	4-16
4.3.2 Simulation Inputs	4-17
4.3.3 Simulation Outputs	4-20

## TABLE OF CONTENTS (Continued)

<u>Section</u>	<u>Page</u>
4.4	4-22
4.4.1	4-22
4.4.2	4-22
4.4.3	4-28
5.	5-1
5.1	5-1
5.2	5-1
5.3	5-2
5.4	5-2
5.5	5-3
5.5.1	5-4
5.5.2	5-5
5.6	5-5
5.6.1	5-5
5.6.2	5-9
5.6.2.1	5-10
5.6.2.2	5-10
5.6.2.3	5-10
5.7	5-15
5.7.1	5-16
5.7.1.1	5-16
5.7.1.2	5-20
5.7.1.3	5-22
5.7.1.4	5-25
5.7.1.5	5-26
5.7.1.6	5-26
5.7.2	5-27
5.7.3	5-27
5.7.4	5-28
5.7.5	5-28
5.7.6	5-29
5.7.7	5-30
5.8	5-30
5.8.1	5-30
5.8.2	5-33
6.	6-1
REFERENCES	6-3
APPENDIX A - PROGRAMS AND FLOW CHARTS	A-1
APPENDIX B - USE OF EXISTING FAST FOURIER TRANSFORM DIGITAL PROCESSOR (FFTDTP) FOR EAR	B-1

## LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
2-1	Dwell Time Limits	2-5
3-1	General Block Diagram of EAR Signal Processor	3-2
4-1	FTS Signal Processor Block Diagram	4-2
4-2	Clutter Map/CFAR Concept	4-7
4-3	Processor Model	4-9
4-4	Probability of Detection: Ideal CFAR	4-24
4-5	S/N Requirements for Detection	4-26
4-6	Relative Response Of 3-Pulse and Narrow-Notch Filters	4-27
4-7	Probability of Detection: Fixed Threshold	4-29
4-8	Clutter Map/CFAR Performance Curves	4-30
4-9	Clutter Map/CFAR Performance Curves	4-31
4-10	Clutter Map/CFAR Performance Curves	4-32
4-11	Clutter Map/CFAR Performance Curves	4-33
5-1	Butterfly Representation	5-6
5-2	Physical Representation of Butterfly	5-7
5-3	Programmable (Pipeline) Signal Processor	5-11
5-4	Distributed Signal Processor	5-12
5-5	Processing Time Versus Transform Size	5-13
5-6	I/O Time Versus Transform Size	5-14
5-7	Signal Processor Block Diagram	5-17
5-8	Signal Processor: GPSP Architecture	5-18
5-9	Distributed Signal Processor	5-19
5-10	Magnitude Determination Flow Chart	5-24
5-11	Cost Comparison: Hardware Only	5-31
5-12	Performance Available for Constant Hardware Cost	5-32
5-13	Cost Comparison: Hardware Plus Development Costs	5-34
5-14	Performance Available for Constant Total Cost ( $C = 200$ )	5-35



## SECTION 1. INTRODUCTION

### 1.1 Purpose of the Study

The broad objective of the work described here has been to conduct a design study aimed at developing and evaluating techniques for suppressing false alarms in air-defense radars. More specifically, the output of the study should define, to the extent allowed by the level of effort expended, circuitry required in a False Target Suppression (FTS) signal processor that would implement these techniques. This processor would be compatible with the U.S. Army Experimental Array Radar (EAR) which is presently installed at the U.S. Army Missile Research, Development and Engineering Laboratory at Redstone Arsenal, Alabama.

### 1.2 Organization of the Study

Pursuant to the above objective, the work (as well as this report) has been organized into the following categories:

#### 1.2.1 Background

The characteristics of the EAR and of its existing MTI processor have been reviewed. The characteristics of the target and clutter environment within which it can be expected to work have been postulated. The characteristics of other processors which have similar FTS functions, both developed within Raytheon or available in the literature, have been reviewed.

#### 1.2.2 General Processor Configuration

Based on the background information, a general FTS block diagram has been generated. This forms a baseline for analytical performance studies as well as detailed processor architecture studies and the cost/performance tradeoffs which follow.

### 1.2.3 Analysis

A detailed analysis has been carried out of the performance of the FTS processor (basically expressed as target detection probability for a fixed false alarm rate) as a function of certain parameters such as number of target samples coherently processed, input signal-to-clutter ratio, input signal-to-noise ratio, presence or absence of MTI canceller processing, etc. This study, although by no means exhaustive, does clearly indicate the performance improvements that might be anticipated by the use of a properly configured FTS processor.

### 1.2.4 Hardware/Software Configurations and Tradeoffs

Carrying out in more detail the work outlined in 1.2.2 above, this task examines various alternative approaches to the design of the processor. More detailed block diagrams are generated, and estimates are made of such cost-influencing factors as the number of components/cards required by each approach and the software requirements, as a function of system specifications such as number of instrumented range cells and number of samples coherently processed.

The information generated in 1.2.4 is combined with rough estimates of relative development costs to form curves of the cost of various design approaches as a function of required performance, or, conversely, the performance that could be expected for a given fixed cost investment. This latter is perhaps the first step toward the application of "design-to-cost" philosophies to signal processor design.

## SECTION 2. BACKGROUND AND ASSUMPTIONS

### 2.1 The Experimental Array Radar (EAR)

The EAR is a C-band phased array radar designed to fulfill both search and track functions in air defense applications (References 1 and 2). Two crossed beams--vertical and horizontal fans--are generated and steered from two linear arrays. False alarm suppression is of primary significance only in the search function, so that this is the basic function considered in this study, although certain of the processing elements described here such as the doppler processor are also of considerable value in improving tracking performance.

For search, only the vertical fan beam, which scans in azimuth over approximately  $\pm 50^\circ$ , is used. Those radar characteristics which are important to the signal processor are listed:

- a. 0.2 microsecond effective pulse width.
- b. Nominal PRF of 5 kHz. This is however variable, and in fact, is almost sure to be varied in a programmed fashion to overcome target blind speed problems and/or as an ECCM measure. We have assumed 5 kHz as an upper limit on the PRF. This implies a maximum of 1000 range cells per PRF interval and a 30 km maximum range, although in practice a smaller number is likely to be instrumented--about 700--in order to allow some radar dead time for uncompressed pulse length (if pulse compression is used), beam steering, etc.
- c.  $2^\circ$  azimuth beamwidth, implying 50 azimuth angle resolution cells per scan. The nominal scan time is assumed to be about 0.5 second (or 10 milliseconds per angle dwell), but it should be stressed that the EAR is an experimental radar so that the ability to vary such parameters is an important part of the design of any processor.

d. There is a 3-pulse digital MTI presently implemented with the EAR. This is a non-moving-window design with one output per three pulses input. The FTS processor should be designed so that some appropriate elements of it can interface with this MTI.

## 2.2 Limits on the Processor Design

As mentioned above, variability is an important property of the FTS processor to be designed. But to keep the design within some limits we must consider certain inherent limits on the radar's output, which are functions of an assumed target environment.

Assume that targets have a maximum velocity of Mach 3 (about 1000 m/sec). The minimum velocity is to be as low as possible. The clutter environment, discussed in a later section is typically, fields, wooded hills, weather, etc.

The antenna scan itself poses no limits on dwell time (and therefore pulses to be processed within one beamwidth). Considered here are other limitations, which apply largely to high-speed targets which in a tactical radar are likely to represent the greatest threat. These are:

- Scan time: The rate at which new information is needed.
- Range walk: The motion of the target through a range cell during the time on target.
- Target acceleration effects: A limit on coherent processing.
- Maximum target coherence time.

### 2.2.1 Scan Time Limitation

The nominal scan time is 0.5 second, corresponding to a dwell time of 10 msec (for  $2^\circ$  beamwidth,  $100^\circ$  search sector). A more generous limit could be established by postulating that a Mach 3 target be detected after penetrating not more than 5 km into the radar coverage: this leads to a scan time of 5 seconds, and a dwell time of 100 msec. This latter is probably an upper bound since it allows no time for radar track functions.

### 2.2.2 Range Walk

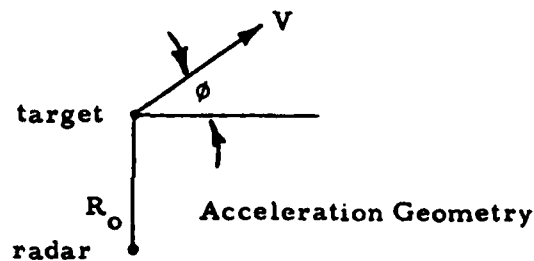
Given a target radial velocity of  $V_R$  m/sec., a target will pass through the 0.2  $\mu$ second (30 m) range cell width in  $T_{RW} = 30/V_R$  seconds. Beyond this limit there is no integration gain within a range cell, although there still will be an increase in overall target detectability due to detection opportunity on adjacent range cells. For a Mach 3 target, this limit is 30 milliseconds (150 pulses at the nominal 5 kHz PRF).

### 2.2.3 Target Acceleration

Here we assume that the target is moving in a straight line at constant velocity: the range acceleration is due to the rate of velocity change for such targets on near-tangential courses. It is easy to show (see the sketch) that the radial acceleration

$\frac{\Delta V_R}{\Delta t}$  is given by:

$$\frac{\Delta V_R}{\Delta t} = \frac{V^2 \cos^2 \phi}{R_o}$$



Where  $V$  is the target velocity,  $R_o$  is the range,  $\phi$  is the off-tangent target angle,  $\Delta t$  is the dwell time, and  $\Delta V_R$  is the change in radial velocity during the dwell time.

Coherent processing will not be efficient if  $\Delta V_R$  equals or exceeds the width of a "doppler bin" (measured in the same units) of the doppler processor. The doppler bin width  $f_B$  for full coherent processing during the time  $\Delta t$  is  $1/\Delta t$  Hz. The equivalent velocity bin width is:

$$V_B = \frac{\lambda f_B}{2} = \frac{\lambda}{2 \Delta t}$$

Then, for  $\phi$  near zero:

$$\Delta V_R = \frac{V^2 \Delta t}{R_o} < V_B, \text{ or } \Delta t < \frac{1}{V} \sqrt{\frac{\lambda R_o}{2}}$$

For  $\lambda = 5.5 \times 10^{-2}$  m,  $V = 10^3$  m/sec., and  $R_o$  (minimum range) =  $10^3$  m,  $\Delta t$  should not exceed 5.2 msec., or the number of pulses coherently integrated (N) need not exceed 26. At maximum 30 km range,  $\Delta t$  is 28.5 msec,  $N = 142$ .

#### 2.2.4 Target Coherence

From Reference 3, page 397, the doppler spreading due to target rotation is given by:

$$f_{MAX} = \frac{2 \omega_r L}{\lambda}$$

Where  $\omega_r$  is the target's real or apparent rotation rate and  $L$  is the target length. For a crossing target as before, the apparent target rotation rate is given by:

$$\omega_r = \frac{V}{R_o} \cos \phi$$

Then  $f_{MAX} = \frac{2LV}{R_o \lambda}$  for a near tangential target.

For  $V = 10^3$  m/sec, a target length of 5 meters, and  $R_o = 10^3$  meters as before,

$$f_{MAX} = 180 \text{ Hz.}$$

The corresponding useful period of target coherence is  $\Delta t = \frac{1}{f_{MAX}} = 5.5$  msec, or 27 pulses. At maximum range,  $\Delta t = 165$  msec.,  $N = 825$ .

The simple relationships derived in sections 2.2.3 and 2.2.4 are plotted in Figure 2-1. These should not be interpreted as meaning that longer dwell times than those indicated cannot be used, but only that the full signal-to-noise improvement that should be expected from coherent integration will not be achieved beyond this dwell time. Since S/N may not be of importance at short range, the dwell time limits at maximum range are probably the most important ones.

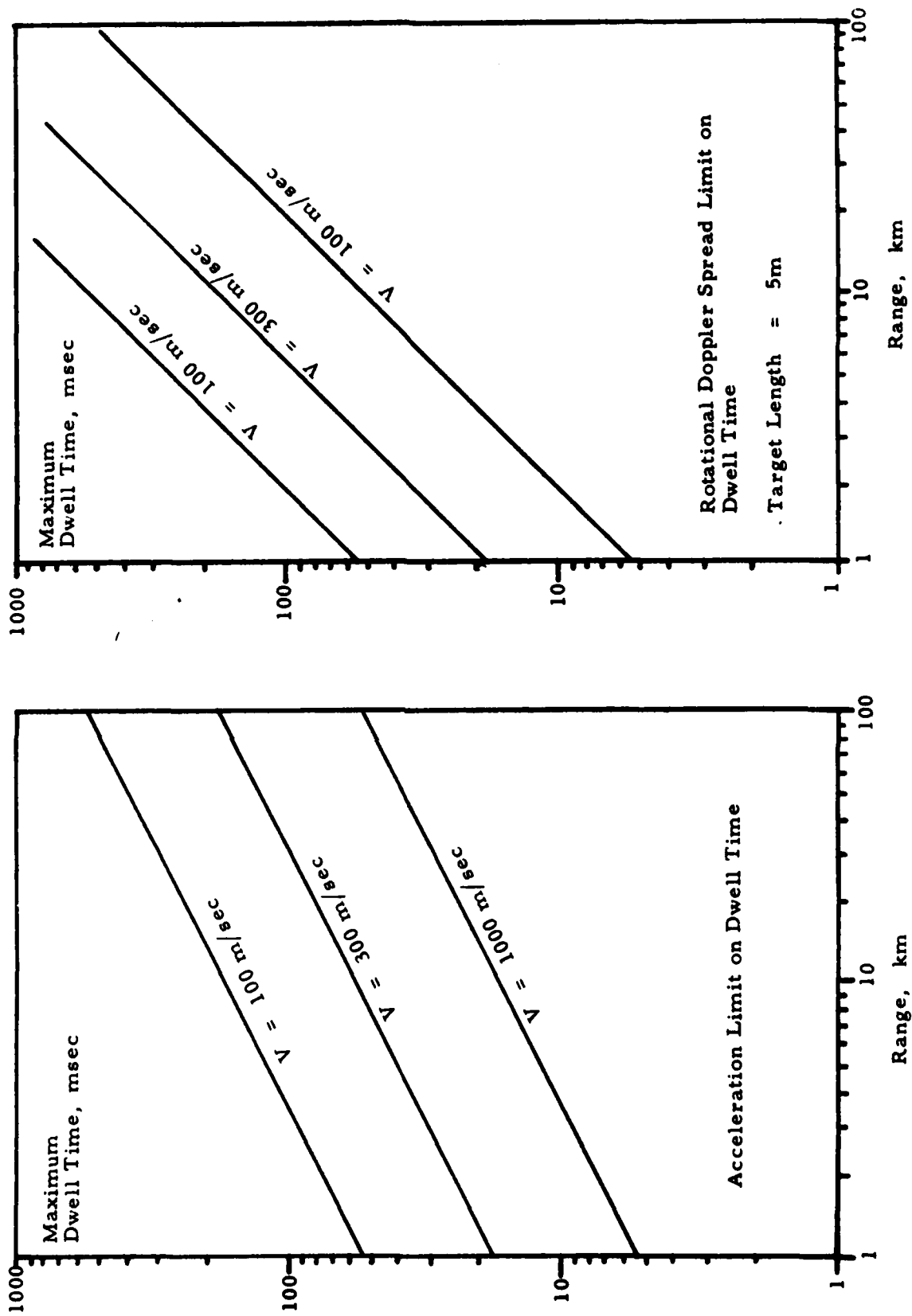


Figure 2-1. Dwell Time Limits

## 2.3 The Clutter Environment

### 2.3.1 Ground Clutter

At the maximum range of 30 km the illuminated ground clutter patch is  $30\text{m} \times 1050\text{m} = 31 \times 10^3 \text{m}^2$  in area. Using C-band data from Reference 4, page 262, the 84th percentile clutter coefficient  $\sigma_o$  is -27 dB for wooded hills. This leads to a clutter cross-section of  $31 \times 10^3 \times .002 = 62\text{m}^2$ . Assuming the smallest target size to be  $0.1\text{m}^2$ , the distributed C/S ratio is about 30 dB.

This neglects the effect of point clutter, which can be considerable. Cross-sections of  $10^4 \text{m}^2$  for point clutter are not unusual (Reference 5). This implies a C/S ratio of 50 dB at these points. Since there are comparatively few of such points, they are a logical candidate for removal by clutter mapping.

The spectrum of the distributed ground clutter (from Reference 4, page 274) is about 0.5 m/sec. in width ( $\pm \sigma$ ) under very strong (~50 kt) wind conditions, proportionately less for lighter winds. At C-band this corresponds to a doppler band of 18 Hz. This is actually  $\pm 9$  Hz, centered at zero frequency.

### 2.3.2 Precipitation Clutter

The volume of rain illuminated is equal to the ground patch area of  $31 \times 10^3 \text{m}^2$  times the height of the rainstorm or the antenna vertically-illuminated altitude, whichever is least. For an assumed storm height of 3 km, the first condition governs, and the volume is about  $10^8 \text{m}^3$  at maximum range.

The backscatter cross-section of the rain is the reflectivity coefficient at C-band times the volume. The reflectivity coefficient  $\mu$ , from Reference 3, page 106, is tabulated below for various rainfall rates:

<u>Rain Rate, mm/hr.</u>		<u><math>\mu, \text{m}^2/\text{m}^3</math></u>	<u><math>\sigma_{\text{rain}}</math> at max. range</u>
Light	1	$5 \times 10^{-9}$	$0.5\text{m}^2$
Moderate	4	$7 \times 10^{-8}$	7.0
Moderate	10	$2 \times 10^{-7}$	20
Heavy	20	$10^{-6}$	100



Typically, (Reference 4, page 211) the rain clutter has a spectral width of up to 5 m/sec. about a center velocity of  $\pm 30$  m/sec., depending of course on wind velocity and antenna direction. At C-band the doppler width is about 180 Hz.

### SECTION 3. THE GENERAL APPROACH

The approach described here is a fully digital processor, both for compatibility with the existing EAR equipment and because analog approaches are not nearly powerful or flexible enough. Several techniques are considered in combination, each reinforcing the other to attack the variety of false target sources which are expected: distributed ground clutter, strong point ground clutter, precipitation clutter, countermeasures and pulsed interference.

The techniques are:

- Doppler processing, as implemented by the Fast Fourier Transform (FFT) technique.
- Range-cell averaging CFAR
- Clutter mapping
- Frequency-cell CFAR
- MTI

Although it is recommended that all of these techniques be implemented to obtain maximum false target suppression with minimum target loss, it is not necessary that they all be implemented in order that benefit be obtained. For example, the range-cell CFAR alone when used with the presently existing EAR will be reasonably effective.

Figure 3-1 illustrates the proposed configuration. The A/D and the lower MTI already exist. The other blocks represent new equipment. Describing in general terms the functions of each block:

#### 3.1 MTI

The existing MTI processes input data in block form producing one output pulse for each three input pulses. It is not suitable as an input to the doppler processor. However it can be input to a cell-averaging CFAR processor, and it is shown as being capable of being switched in at that point.

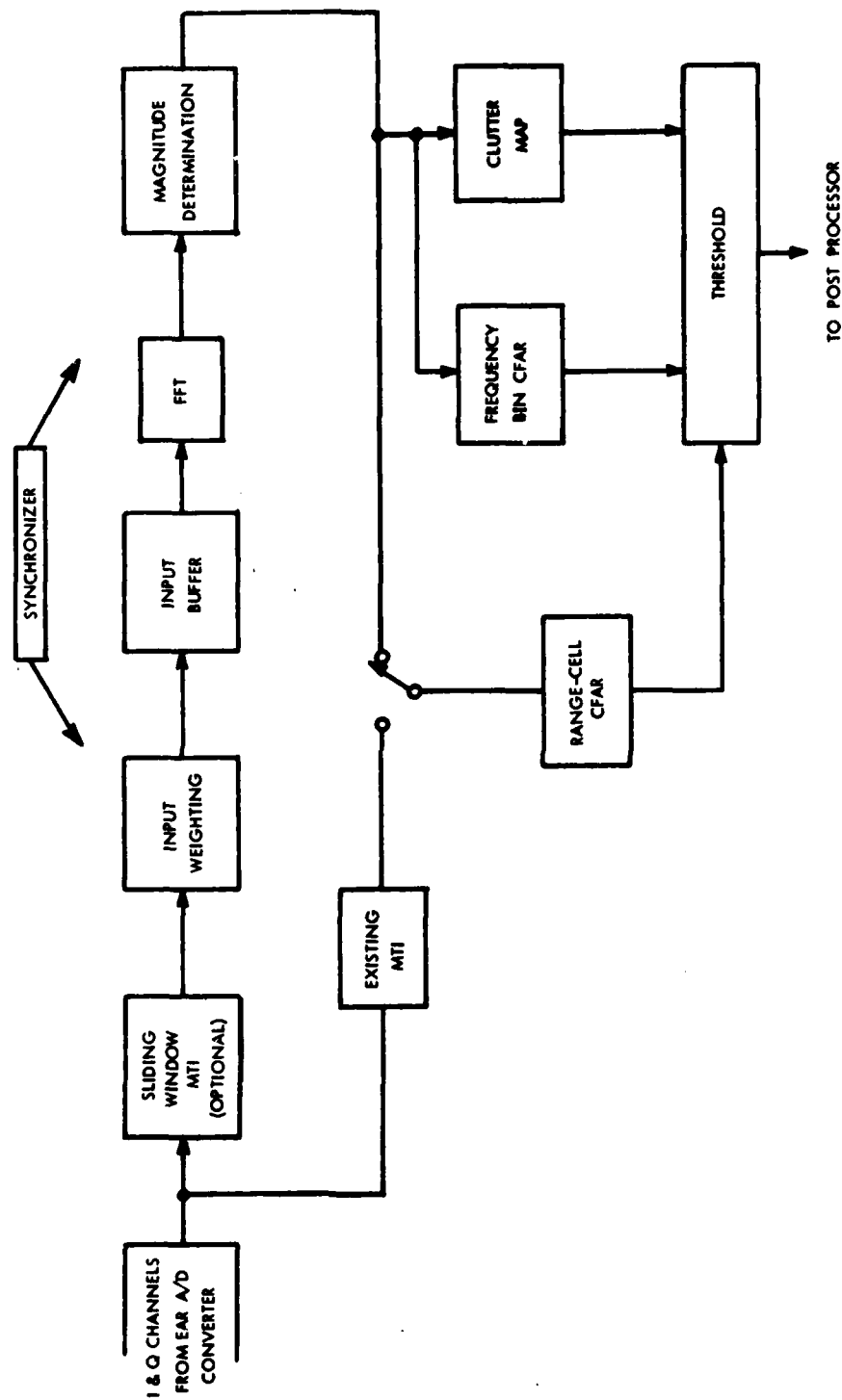


Figure 3-1. General Block Diagram of EAR Signal Processor

The sliding window MTI is suitable for input to the FFT, and is so shown. Although "optional" in the sense that the FFT itself by virtue of its doppler filtering behavior has certain MTI properties, it is shown in Section 4 that the implementation of an MTI at this point will enhance the total performance of the FTS processor in a clutter environment.

### 3.2 Input Weighting

The FFT is a block processor, processing sequential pulse trains which are, most efficiently, powers of two in length. The truncation implicit in the processing of a finite block like this will give rise to  $\sin x/x$  sidelobes in the frequency domain, with the worst sidelobe some 13 dB down. By pre-multiplying the input pulse sequence by some sort of a smooth amplitude weighting function, the sidelobe level can be controlled to any desired degree, with a price being paid in widening the bandwidth of the doppler filter and in reduced signal-to-noise ratio. The Taylor weighting function is the most efficient and is the one considered here.

### 3.3 Input Buffer

This is actually a double buffer. A complete block of input data (N pulses by M range cells) is fed in real time into one buffer while the FFT is processing the contents of the other buffer, which had been filled with data earlier. The roles of the two buffers are reversed when the real time buffer is filled.

### 3.4 The Fast-Fourier Transform (FFT) Processor

The principles of operation of the FFT are well described in the literature. Given a time series input N complex (I & Q) samples in length, the FFT produces an N point spectrum analyzer output.

### 3.5 Magnitude Determination

The FFT spectrum output is complex, comprised of I and Q components. Only the magnitude is of interest to us. The magnitude unit computes a good approximation of the magnitude  $\sqrt{I^2 + Q^2}$  for each spectrum point (filter).

### 3.6 Range-Cell CFAR

This technique is well-described in the literature. See References (6) and (7). Each FFT filter output except the zero frequency one has its own range-cell CFAR processor which produces a threshold which applies to that filter only.

### 3.7 Frequency-Bin CFAR

This processor sets a threshold for each range cell which is derived from the average of all filter outputs except those around zero frequency. As a result any signal whose energy is spread more or less uniformly throughout the spectrum instead of being concentrated in one or two frequency bins is excluded. This gets rid of certain kinds of pulse interference and jamming.

### 3.8 Clutter Map

Really strong point clutter can break through the MTI/FFT process. The clutter map suppresses this residue by storing the locations of these clutter points (which appear in the zero frequency filter) and setting a threshold in the zero bin and proportionately in near-zero bins.

### 3.9 Threshold

We have described three sources of threshold in 3.6, 3.7 and 3.8 above. The signal must exceed each of these thresholds if it is to be output to the post processor--essentially a logical "and".

## SECTION 4. ANALYTICAL FTS STUDY

### 4.1 Introduction

This section reports on a clutter map/CFAR study aimed at evaluating the performance of several candidate false target suppression (FTS) signal processors for the Experimental Array Radar (EAR). In 4.2, a theoretical model is developed for comparing the performance of several digital doppler processors in realistic clutter environments. This model is implemented in a computer simulation described in Section 4.3, and performance curves (probability of detection) are given in Section 4.4 for each of the candidate processors and different signal-to-interference (S/I) conditions. From these curves, the advantages of including the MTI and appropriate pre-FFT weighting are clear for heavy clutter environments. With MTI there will, however, be a tendency to lose low-velocity targets or those at or near blind velocities. There exist techniques (range-gated MTI, staggered prf) which can minimize these effects.

The basic components of the signal processor are given in Figure 4-1. The system begins by sampling the inputs from the analog in phase (I) and quadrature (Q) channels of the receiver with an analog-to-digital converter (ADC) of appropriate characteristics. The resulting output is fed through an MTI canceller ahead of the FFT in order to reduce dynamic range and to improve clutter rejection and is then weighted appropriately to reduce doppler sidelobe effects. Since doppler processing is associated with operations on blocks or "batches" of samples, the weighting network output is input into a buffer storage and an FFT then translates these samples from the time domain to the frequency domain. Finally, the FFT outputs are fed into an FTS processor whose function is to reduce the number of clutter false alarms. This is achieved by applying a number of threshold tests to the range gate/filter amplitudes. The clutter map provides a scan-to-scan history of the clutter in the zero velocity doppler filter and defines an adaptive threshold for the low velocity doppler filters. Also, a cell-averaging CFAR processor further eliminates the effects of clutter in the higher velocity doppler channels.

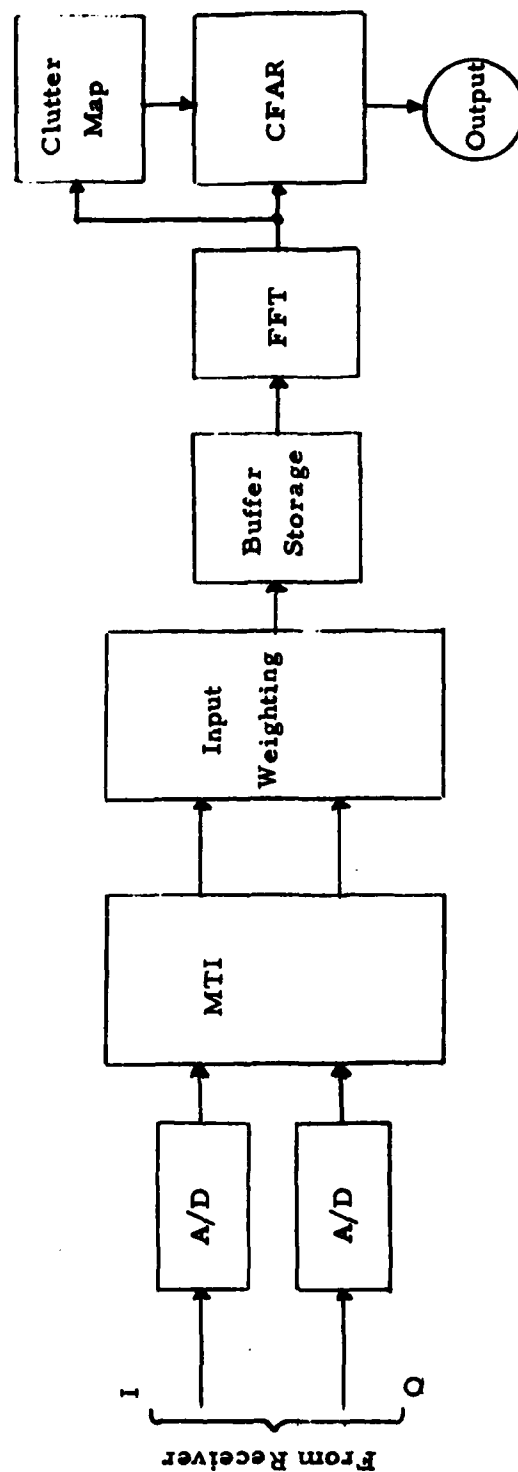


Figure 4-1. FTS Signal Processor Block Diagram

## 4.2 Theoretical Model

In this section, a mathematical model is presented for computing detection probabilities of the output of several clutter map/CFAR (CM/CFAR) signal processors. Specifically, the configurations analyzed in this and later sections are

1. ADC  $\rightarrow$  MTI  $\rightarrow$  W  $\rightarrow$  FFT  $\rightarrow$  CM/CFAR
2. ADC  $\rightarrow$  MTI  $\rightarrow$  FFT  $\rightarrow$  CM/CFAR
3. ADC  $\rightarrow$  W  $\rightarrow$  FFT  $\rightarrow$  CM/CFAR
4. ADC  $\rightarrow$  FFT  $\rightarrow$  CM/CFAR
5. ADC  $\rightarrow$  MTI  $\rightarrow$  CFAR

where W denotes a weighting network. Although several processors are considered, one general math model is derived which covers each of the specific designs with appropriate modification of the signal covariance matrices at the FFT input.

### 4.2.1 Signal Models

Target, ground clutter, weather clutter, and noise signal models needed for evaluating detection performance in realistic clutter (Interference) environments are given below:

#### a. Target Signal.

The target model appropriate for the step-scanned EAR radar is the Swerling I model. Here the target signal power returned per pulse is assumed to be constant for the time on target during a single scan, but to fluctuate independently from scan-to-scan. Expressed in statistical terms, the normalized autocorrelation function of target cross section is approximately one for the time in which the beam is on target during a single scan and is approximately zero for a time as long as the interval between scans.

#### b. Ground Clutter Model.

As described in Barton [3], ground clutter consists of a fixed component due to returns from rigid scattering elements within the clutter patch (e. g. buildings, fences, power poles, etc. ) plus a fluctuating component due to returns from moving reflectors such as swaying trees, vegetation, etc. The clutter voltage samples for a given range bin are therefore in general characterized temporarily by a Rician amplitude distribution. In this work, however, the fixed (DC) component



of clutter is assumed absent. This assumption is reasonable for processors containing an MTI, and should give reliable detection probabilities for the case of point clutter and no MTI in all but the first few doppler bins. Thus, in the comparisons between systems with and without MTI, the detection probabilities for the NO-MTI case may be somewhat optimistic for the first few FFT filters where the given detection performance should be considered only as an upper bound to the actual system performance. With this in mind, the ground clutter voltage samples for a given range bin are assumed to be characterized temporally by a Rayleigh amplitude distribution with density function

$$p(z) = \frac{z}{\sigma^2} e^{-z^2/2\sigma^2} \quad (4-1)$$

Here  $P_c = 2\sigma^2$  is the average clutter power from a given range bin and is proportional to the clutter backscatter coefficient,  $\sigma^0$ .

As described in Boothe [8], since the clutter backscatter coefficient varies from clutter patch to clutter patch,  $\sigma^0$  is itself a random variable. We assume here that  $\sigma^0$  is a Weibull variable with the probability density function

$$p(\sigma^0) = \frac{b}{\alpha} (\sigma^0)^{b-1} \exp \left( - \frac{(\sigma^0)^b}{\alpha} \right) \quad (4-2)$$

where

$$b = \frac{1}{A} \quad (A = \text{Weibull slope parameter} = 2)$$

$$\alpha = \left[ \frac{\bar{\sigma}^0}{\Gamma(1+A)} \right]^b$$

and  $\bar{\sigma}^0$  is the backscatter coefficient averaged over the CFAR window. In reference [8], Boothe shows that this Weibull assumption is reasonable over land and gives typical values of the slope parameter A. For semi-wooded, hilly terrain,  $A = 2$  is representative of the average clutter environment and is therefore used in this analysis.

Nathanson [4] shows typical land clutter profiles versus range obtained by averaging clutter returns from many pulses. These clutter profiles were analyzed by Brooks [9] and were also found to fit reasonably well with

a Weibull distribution having an exponential range bin-to-range bin correlation. Equivalently, clutter power versus range is not constant but exhibits the same spatial variation as that of the clutter backscatter coefficient. Reference [10] suggests a strong range bin-to-range bin correlation for land clutter (i. e.,  $\rho > .99$ ). In this analysis, we have selected the correlation  $\rho = 1$  because it is mathematically easy to evaluate CFAR detection performance in this case and the performance for  $\rho > .99$  should be very nearly identical to the performance with  $\rho = 1$ . This assumption is equivalent to assuming that the average clutter plus noise power is equal in each range bin of the CFAR window.

Finally, we assume the pulse-to-pulse correlation between ground clutter returns is given by the correlation function

$$R(\tau) = e^{-(2\pi\sigma_c\tau)^2/2} \quad (1) \quad (4-3)$$

where  $\sigma_c$  is the rms frequency spread of the clutter and is related to the rms velocity spread of the scattering elements  $\sigma_v$  by the familiar doppler equation

$$\sigma_c = \frac{2\sigma_v}{\lambda} \quad (4-4)$$

At C-Band ( $\lambda = .0545M$ ), the spectral width of ground clutter can range from about .6 Hz on a calm day to about 18 Hz for winds greater than 40 knots.

#### c. Weather Clutter Model

The weather clutter is assumed to have a Rayleigh temporal distribution with the pulse-to-pulse correlation

$$R(\tau) = \exp \left[ - (2\pi\sigma_w\tau)^2/2 \right] \cos (2\pi f_w\tau) \quad (4-5)$$

where  $\sigma_w$ , analogous to  $\sigma_c$ , is the spectral width of the rainstorm which at C-Band ranges from about 67 Hz ( $\sim 6$  ft/sec) for light rain to about 145 Hz ( $\sim 13$  ft/sec) for heavy rain [3]. Also,  $f_w$  denotes the center frequency of the storm and can range

---

(1) This correlation function is consistent with the assumption of a Gaussian clutter spectrum.

from 0 Hz to about 945 Hz ( $\sim 50$  knots). It should be pointed out that the correlation function (4-5) is equivalent to the assumption of a Gaussian spectrum for the case of zero center frequency,  $f_w$ , and although it deviates from this assumption for non-zero center frequency, the associated spectrum will approximate a Gaussian spectrum for frequencies between zero and  $1/2$  PRF (2.5 kHz) when the center frequency is large compared to the rms frequency spread of the storm. These two extreme cases (center frequencies zero and 945 Hz) are the two cases analyzed in Section 4.3.

Weather clutter will generally be dispersed in range over several cells. In this analysis, the mean clutter powers over the CFAR windows to the left and to the right of the target cell are assumed constant but not necessarily equal.

d. Noise Model

The noise model used is characterized by a Rayleigh amplitude distribution with zero pulse-to-pulse correlation (white-noise).

4.2.2 Clutter Map/CFAR Concept

The basic clutter map/CFAR concept used here for a FTS doppler processor is illustrated in Figure 4-2. The range bin of interest containing the target is depicted as  $R_0$  and the  $N$  range cells to the left and right of  $R_0$  are shown as  $R_{-1}, \dots, R_{-N}$  and  $R_1, \dots, R_N$  respectively. An  $L$ -point FFT (filters denoted 0 through  $L-1$ ) is taken of the signals from each of the range bins. In the figure, from the FFT on, the flow is shown for target detection in the  $k$ th filter (and  $R_0$  range bin) only but a similar flow is assumed for each of the remaining filters (and range bins) except for the zeroth filter where only a clutter map threshold is established. The basic idea is to establish appropriate thresholds, compare the  $k$ th filter output of range bin  $R_0$  with the largest threshold determined, and to declare detection if the signal exceeds this maximum threshold. There are three thresholds established for each non-zero filter in the processors studied here, two CFAR (constant false alarm rate) thresholds and a clutter map related threshold. (A fourth threshold derived from frequency bin averaging - see 3.7 - is not included in this analysis, since it is only effective under conditions of pulse interference which are not included in the input signal model.)

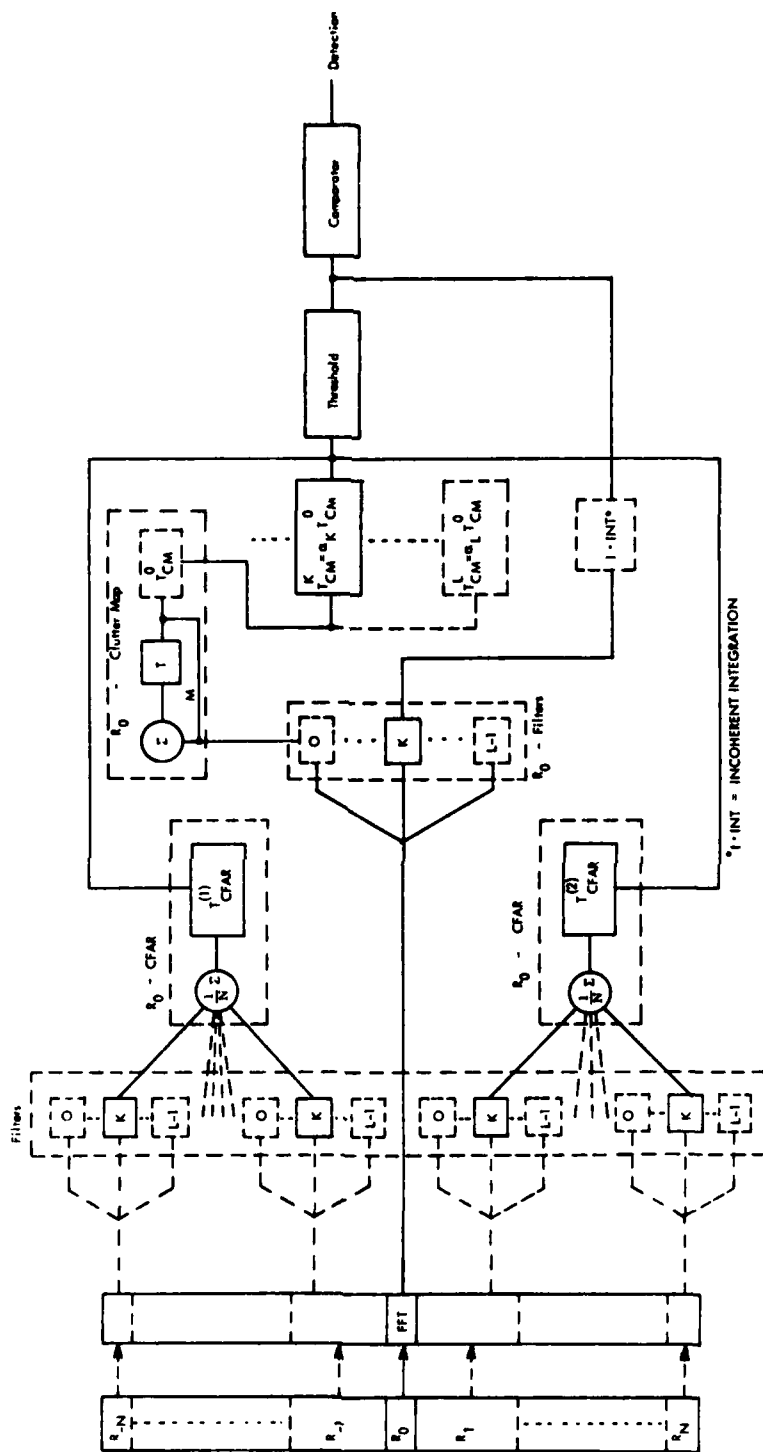


Figure 4-2. Clutter Map/CFAR Concept

The CFAR thresholds  $T_{CFAR}^{(1)}$ ,  $T_{CFAR}^{(2)}$  are simply weighted averages<sup>(1)</sup> of the  $k$ th filter outputs of  $N$  range bins to the left and right of  $R_0$ , respectively. These two thresholds are intended to regulate false alarms in a noise plus weather environment. The leading and lagging cells are averaged separately to establish thresholds that will regulate false alarms in a non-stationary background, as would exist, for example, when the window confronts the edge of a weather mass.

A clutter map threshold  $T_{CM}^0$  is established for the zeroth FFT filter (each range bin) to regulate ground clutter false alarms, and represents a weighted value of the clutter level averaged over several past radar scans. A clutter map-related threshold  $T_{CM}^{(K)}$  is established for the  $k$ th filter and is a fraction  $\alpha_K$  of the  $T_{CM}^0$  threshold. Here  $\alpha_K$  is variable and is programmed into the processor according to the expected doppler roll-off of the ground clutter.

#### 4. 2. 3 Detection Probability-Model

##### a. Interference Distribution At FFT Output

We begin by establishing the noise plus ground clutter plus weather clutter signal distribution for a given range bin  $R_0$  at the  $k$ th FFT filter output Point (B) of Figure 4-3. To do this, we make an assumption on the initial distributions of these individual interference signals at the output of the A/D converters - Point (A). Consistent with the interference models of Section 4. 2. 1, each of these initial signals has a Rayleigh amplitude distribution, that is, the associated in phase (I) and quadrature phase (Q) signals are independent Gaussian variables with zero mean and appropriate pulse-to-pulse correlation, the degree of which depends on the type of interference. Accordingly, we can develop the  $k$ th FFT filter interference statistics by starting with a Gaussian distribution (representing any one of the interference signals) in say, the I-channel, following this distribution through the MTI, weighting and FFT networks, and finally combining the resulting Gaussian distribution with an equivalent Q-channel distribution.

Let the initial I-channel distribution be described by a zero-mean Gaussian distribution with joint density -

---

(1) The weighting applied is consistent with a  $10^{-6}$  probability of false alarm (PFA).

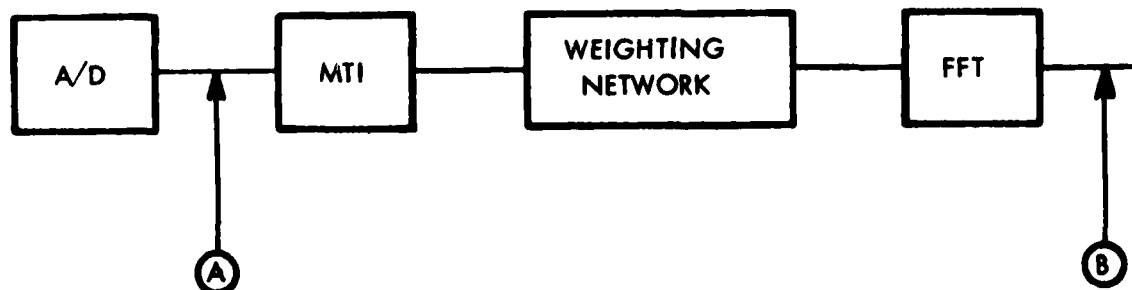


Figure 4-3. Processor Model

$$p(\bar{v}) = \left( \frac{1}{(2\pi)^{n/2} |M_n|^{1/2}} \right) \exp \left( -\frac{1}{2} \bar{v}^T M_n^{-1} \bar{v} \right) \quad (4-6)$$

Here

$$\bar{v}^T = (v_1, v_2, \dots, v_n)$$

is the transpose of an  $n$ th dimensional vector whose elements  $v_i, i = 1, \dots, n$  denote I-channel voltages associated with the first  $n$  pulses, respectively, and  $|M_n|$  is the determinant of the  $n \times n$  covariance matrix  $M_n$  describing the pulse-to-pulse correlation. For a sliding window 3-pulse MTI and the inputs  $v_1, \dots, v_n$ , the  $n-2$  outputs  $\xi_i = v_i - 2v_{i+1} + v_{i+2}, i = 1, \dots, n-2$  again have a zero mean normal distribution with joint density function given by (4-6) with  $\bar{v}$  replaced by  $\xi$  and  $M_n$  replaced by  $M_{n-2}^{(1)} = A M_{n-2} A^{-1}$ , where  $M_{n-2}$  is the matrix  $M_n$  with the last two rows and columns deleted and  $A$  is the linear transformation matrix defined by

$$\xi = A \bar{v} \quad (4-7)$$

$$A = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \dots & 1 & -2 & 1 \end{bmatrix}_{n-2 \times n-2} \quad (4-8)$$

If a weighting network is included and  $C_1, \dots, C_{n-2}$  denote the pre-FFT weights (Taylor), the covariance matrix  $M_{n-2}^{(2)}$  for the zero mean I-channel Gaussian distribution at the FFT input is given by

$$M_{n-2}^{(2)} = C M_{n-2}^{(1)} C^{-1} \quad (4-9)$$

where

$$C = \begin{bmatrix} C_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & C_{n-2} \end{bmatrix} \quad (4-10)$$

For convenience, let  $L = n-2$  be the number of FFT points,  $W_1, \dots, W_L$  denote the I-channel FFT inputs,  $M_L^{(2)} = [m_{ij}]_{L \times L}$ , and  $I_k$  denote the  $k$ th FFT filter output in the I-channel, then

$$I_k = \sum_{j=1}^L W_j \exp[-i(2\pi/L)(j-1)k], \quad k = 0, 1, \dots, L-1 \quad (4-11)$$

$$R(I_k) - iJ(I_k)$$

where the real and imaginary parts  $R(I_k)$  and  $J(I_k)$  are given by

$$R(I_k) = \sum_{j=1}^L W_j \cos \left[ \frac{2\pi}{L} (j-1)k \right] \quad (4-12)$$

$$J(I_k) = \sum_{j=1}^L W_j \sin \left[ \frac{2\pi}{L} (j-1)k \right] \quad (4-13)$$

Using the notation  $N(\mu, \sigma^2)$  to denote a normal distribution with mean  $\mu$  and variance  $\sigma^2$ , it is clear that

$$\left. \begin{aligned} R(I_k) &\in N(0, \sigma_{Rk}^2) \\ J(I_k) &\in N(0, \sigma_{Jk}^2) \end{aligned} \right\} \quad (4-14)$$

where

$$\left. \begin{aligned} \sigma_{Rk}^2 &= \sum_{i,j=1}^L m_{ij} C_{ki} C_{kj} \\ \sigma_{Jk}^2 &= \sum_{i,j=1}^L m_{ij} d_{ki} d_{kj} \\ C_{kj} &= \cos \left[ \frac{2\pi}{L} (j-1)k \right] \\ d_{kj} &= \sin \left[ \frac{2\pi}{L} (j-1)k \right] \end{aligned} \right\} \quad (4-15)$$

In a similar fashion for the Q-channel distributions

$$\left. \begin{aligned} R(Q_k) &\in N(0, \sigma_{Rk}^2) \\ J(Q_k) &\in N(0, \sigma_{Jk}^2) \end{aligned} \right\} \quad (4-16)$$



where again  $\sigma_{Rk}^2$  and  $\sigma_{Jk}^2$  are given by (4-15). Now combining the I and Q channel distributions, the total complex FFT output at the  $k$ th filter becomes

$$\begin{aligned} F_k &= I_k + iQ_k \\ &= [R(I_k) + J(Q_k)] + i[R(Q_k) - J(I_k)] \\ &= R(F_k) + iJ(F_k) \end{aligned} \quad (4-17)$$

where both  $R(F_k)$  and  $J(F_k) \in N(0, \sigma_{Rk}^2 + \sigma_{Jk}^2)$  and are independent. Finally, assuming a square law magnitude unit after the FFT<sup>(1)</sup>, the combined signal

$$Z_{jk} = R(F_k)^2 + J(F_k)^2 \quad (4-18)$$

for range bin  $j$  and out of the  $k$ th FFT filter has the exponential distribution

$$p(Z_{jk}) = \frac{1}{2\sigma_{jk}^2} \exp \left( -Z_{jk}/2\sigma_{jk}^2 \right) \quad (4-19)$$

where

$$\sigma_{jk}^2 = \sigma_{Rk}^2 + \sigma_{Jk}^2 \quad (4-20)$$

Inasmuch as the distribution (4-6) is representative of each of the three independent interference sources, the distribution (4-19) is also representative of the combination of ground clutter, weather clutter, and noise if

$$\sigma_{jk}^2 = \sigma_{Njk}^2 + \sigma_{gjk}^2 + \sigma_{Njk}^2 \quad (4-21)$$

Here the noise, ground clutter, and weather clutter variances  $\sigma_{Njk}^2$ ,  $\sigma_{gjk}^2$ ,  $\sigma_{Njk}^2$  respectively, are each determined from (4-15) and (4-20) after the computation of the appropriate covariance matrices using the assumed pulse-to-pulse and spatial correlations given in Section 4.2.1

#### 4.2.4 False Alarm Probabilities

For a given threshold voltage  $V_{Tk}$ , the false alarm probability at the  $k$ th filter output and range  $R_0$  is

<sup>(1)</sup> Although a linear magnitude unit is actually proposed, a square law unit is used here to simplify the mathematics. Experience shows that the results for either will be very nearly the same.

$$\begin{aligned} \text{PFA}_k &= \frac{1}{2\sigma_{ok}^2} \int_{V_{Tk}}^{\infty} \exp(-\xi/2\sigma_{ok}^2) d\xi \\ &= \exp(-Y_{bk}) \end{aligned} \quad (4-22)$$

where

$$Y_{bk} = V_{Tk}/2\sigma_{ok}^2 \quad (4-23)$$

is a normalized threshold equal in general to the maximum of several established values, except when  $k = 0$  and it is simply the clutter map threshold. In this case,

$$Y_{bo} = \frac{K_1}{2M\sigma_{oo}^2} \sum_{i=1}^M Z_{oo} \quad (4-24)$$

where the notation  $\sum_{i=1}^M Z_{oo}$  is used to denote a sum of signals from range bin 0 and filter 0 over  $M$ -past radar scans, and  $K_1$  is a clutter map constant associated with a certain false alarm probability. Using (4-19), it may be shown that the probability density for  $Y_{bo}$  is

$$p(Y_{bo}) = \frac{1}{(M-1)!} \left( \frac{M}{K_1} \right)^M Y_{bo}^{M-1} \exp \left( -\frac{MY_{bo}}{K_1} \right) \quad (4-25)$$

Whence, the average clutter map false alarm probability is given by

$$\overline{\text{PFA}}_o = \frac{1}{\left( 1 + \frac{K_1}{M} \right)^M} \quad (4-26)$$

and the value of  $K_1$  corresponding to an average false alarm probability of  $10^{-6}$  is

$$K_1 = M(10^{6/M} - 1) \quad (4-27)$$

For the non-zero filters, a clutter map related threshold is compared with the CFAR thresholds and the largest is chosen for the detection criterion. Denoting the normalized clutter map related threshold by  $A_k$ , we set

$$A_k = V_{Tk} / 2\sigma_{ok}^2 = \alpha_k V_{To} / 2\sigma_{ok}^2 \quad (4-28)$$

$$= \left( \frac{\alpha_k \sigma_{oo}^2}{\sigma_{ok}^2} \right) Y_{bo} = \beta_k Y_{bo}$$

where

$$\alpha_k = (\sigma_{gok} / \sigma_{goo})^2 \quad (4-29)$$

is the expected ground clutter roll-off with frequency. The probability density for  $A_k$  is then given as:

$$p(A_k) = \frac{1}{\beta_k} \left[ \frac{1}{(M-1)!} \right] \left[ \frac{M}{K_1} \right] \left[ \frac{A_k}{\beta_k} \right]^{M-1} \exp \left( - \frac{M A_k}{K_1 \beta_k} \right) \quad (4-30)$$

and the average false alarm probability associated with  $A_k$  is

$$\overline{P}(A_k) = \frac{1}{\left( 1 + \frac{\beta_k K_1}{M} \right)^M} \quad (4-31)$$

With regard to the CFAR thresholds, we let  $B_k$  and  $C_k$  denote normalized thresholds associated with the range bins to the left and right of  $R_o$ , respectively. Then,

$$B_k = \frac{k_2}{2N\sigma_{ok}^2} \sum_{j=-1}^{-N} Z_{j,k} \quad (4-32)$$

$$C_k = \frac{k_2}{2N\sigma_{ok}^2} \sum_{j=1}^N Z_{j,k} \quad (4-33)$$

where the sums are over the  $N$  range bins to the left and right of  $R_o$ , respectively, the normalization is with respect to the average interference power from  $R_o$ , and  $k_2$  is a CFAR constant associated with a given false alarm probability. It is now assumed that the average interference powers are constant over the cells to either side of  $R_o$ , but not necessarily equal from side to side. Accordingly, the average

interference power on each cell to the left of  $R_0$  is denoted  $2\sigma_{-1,k}^2$ , and to the right of  $R_0$ , the notation  $2\sigma_{1,k}^2$  is used. Paralleling (4-30), the probability densities for  $B_k$  and  $C_k$  are given by

$$p(\xi_k) = \frac{1}{\tau_k(N-1)!} \left(\frac{N}{k_2}\right)^N \left(\frac{\xi_k}{\tau_k}\right)^{N-1} \exp\left(-\frac{N}{k\tau_k} \xi_k\right) \quad (4-34)$$

where

$$\tau_k = \begin{cases} \sigma_{-1,k}^2 / \sigma_{0,k}^2 & \text{if } \xi_k = B_k \\ \sigma_{1,k}^2 / \sigma_{0,k}^2 & \text{if } \xi_k = C_k \end{cases} \quad (4-35)$$

and the associated false alarm probabilities are

$$\overline{\text{PFA}}_k = \frac{1}{\left(1 + \frac{\tau_k k_2}{N}\right)^N} \quad (4-36)$$

Here  $k_2$  is determined assuming  $\sigma_{-1,k}^2 = \sigma_{1,k}^2$  and is therefore given by

$$k_2 = N(10^{-6/N} - 1) \sigma_{0,k}^2 / \sigma_{1,k}^2 \quad (4-37)$$

Finally, on the average, the  $k$ th filter threshold  $Y_{bk}$  is given as

$$Y_{bk} = \max(A_k, B_k, C_k) \quad (4-38)$$

and

$$\overline{\text{PFA}}_k = \begin{cases} \frac{1}{\left(1 + \frac{\tau_k k_1}{M}\right)^M} & \text{if } Y_{bk} = A_k \\ \frac{1}{\left(1 + \frac{\tau_k k_2}{N}\right)^N} & \text{otherwise} \end{cases} \quad (4-39)$$

#### c. Target Detection Probability - Swerling I Target

With the total average interference power  $2\sigma_{0,k}^2$  known for each filter  $k$ , the average signal-to-interference powers  $\bar{x}_0, \bar{x}_1, \dots, \bar{x}_{L-1}$  vs frequency out of the magnitude network are determined by first passing a sinusoid of unit peak

amplitude (representing a normalized fixed amplitude target signal over the  $L + 2$  pulses) through a 3-pulse MTI and weighting network, if used, and then through an  $L$ -point FFT and finally through the magnitude unit. Denoting these samples by  $s_0, \dots, s_{L-1}$ , the average signal-to-interference samples vs. frequency for a Swerling I target are given as:

$$\bar{x}_k = \left( \frac{S}{I} \right) \left( \frac{S_k}{\hat{\sigma}_{o,k}^2} \right) \quad (4-40)$$

where  $S/I$  is the average signal-to-interference before the MTI;

$$\hat{\sigma}_{o,k}^2 = \frac{\sigma_{o,k}^2}{\sigma_I^2}, \text{ and } 2\sigma_I^2 \text{ is the total interference power before the MTI.}$$

The detection probabilities for a single pulse out of the magnitude unit are then given by the well known Swerling formula

$$P_{Dk} = \exp(-Y_{bk}/(1 + \bar{x}_k)) \quad (4-41)$$

and the average detection probabilities are

$$P_{Dk} = \begin{cases} \left( \frac{1}{1 + \frac{B_k k_1}{M(1 + \bar{x})}} \right)^M & \text{if } Y_{bk} = A_k \\ \left( \frac{1}{1 + \frac{\tau_k k_2}{N(1 + \bar{x})}} \right)^N & \text{otherwise} \end{cases} \quad (4-42)$$

As remarked earlier, the average ground clutter powers  $2\sigma_{gk}$  are, for each  $k$ , constant over the CFAR window, but are nevertheless random and proportional to the Weibull distributed backscatter coefficient  $\sigma^0$ . Averaging (4-42) over the Weibull distribution gives the overall average detection probability

$$\langle P_{Dk} \rangle = \int_0^\infty \bar{P}_{Dk}(\sigma^0) p(\sigma^0) d\sigma^0 \quad (4-43)$$

### 4.3 Clutter Map/CFAR Computer Simulation

#### 4.3.1 Introduction

A simulation has been developed for computing detection probabilities according to the model given in Section 4.2. The simulation is divided into two separate

programs, CFAR and CFARP. A top level flow diagram and listings of each of these programs is given in Appendix A.

The program CFAR generates the initial interference covariance matrices and performs the linear transformations necessary for establishing the mean interference powers for each FFT filter at the output of the magnitude unit. These interference signals are stored on direct access file as a function of frequency (filter #) and are automatically called for by the program CFARP, which in turn computes target detection probabilities as a function of target doppler frequency for various S/I conditions.

#### 4.3.2 Simulation Inputs

The operator inputs for the programs CFAR and CFARP are given in Tables 4-1 and 4-2 below.

Table 4-1. CFAR Inputs

Variable Name	Definition
A	Weibull slope parameter.
ATAY	Sidelobe level control parameter for Taylor weights.
FW	Center frequency for rain storm (Hz).
GM	Logical variable - Turns ground clutter model on or off. This variable should always be set = . True. on input.
IWGT	Logical variable - IWGT = T for Taylor weighting; IWGT = F for no pre-FFT weighting.
KCLUT	Logical variable - KCLUT = T results in a call exit since a D. C. ground clutter component has not yet been programmed into the simulation.
MTI	Logical variable - MTI = T for an MTL, and MTI = F for no MTL.
MTIMDL	MTIMDL = 1 for 3 pulse sliding window MTI; MTIMDL = 2 for narrow notch MTL.
MUW0	Backscatter cross section of the rain ( $M^2$ ) - target range cell.
MUW1	Backscatter cross section of the rain ( $M^2$ ) - to the left of target cell.
MUW2	Backscatter cross section of the rain to the right of target cell.

Table 4-1. CFAR Inputs (Continued)

<u>Variable Name</u>	<u>Definition</u>
NBAR	Number of terms used in Taylor weighting function - used only in the case of pre- FFT weighting
NM	Logical variable - Set NM = T in all runs.
NPTS	Number of FFT points (maximum of 64).
SIGSG	Spectral width of ground clutter (Hz).
SIGSW	Spectral width of weather clutter (Hz).
STGCDB	Signal-to-ground clutter ratio (dB).
STKCLDB	Signal-to-point clutter ratio (dB) - not used in present form of the simulation since a point clutter model has not yet been added to the program.
STNDB	Signal-to-noise ratio (dB).
THMDL	Threshold model. Set THMDL = 1 in all computer runs.
WM	Logical variable. WM = T for rain; WM = F for no rain.
WMDL	Weather model used; WMDL = 1 in all computer runs.

Table 4-2. CFARP Inputs

<u>Variable Name</u>	<u>Definition</u>
IATTN	Logical variable. IATTN = T for a plot of ground clutter plus noise attenuation vs. frequency.
IDEALCF	Logical variable. IDEALCF = T if a plot of detection probability is desired for a fixed threshold.
IFRESP	Logical variable. IFRESP = T for a plot of MTI response vs. frequency. The plots are normalized to the noise out of the 0th filter.
IPROB	Logical variable. IPROB = T for Plots on Probability Paper.
ITATTN	Logical variable. ITATTN = T for a plot of clutter map threshold attenuation vs. frequency.
IWGT	See CFAR Inputs.
MORCFAR	Logical variable. Set MORCFAR = F if only the ITATTN or IATTN plots are desired as output.

Table 4-2. CFARP Inputs (Continued)

Variable Name	Definition
MTI	See CFAR.
MTINDL	See CFAR.
NCLMAP	Age of clutter map in number of interference samples averaged.
NRCELLS	Total number of range cells in cell averaging CFAR NRCELLS/2 cells to each side of target cell).
PRED	Logical variable. Set PRED = T if predicted clutter map/CFAR detection curves are desired.
PRFA	Logical variable. Set PRFA = T if false alarm probability at each FFT filter is desired for a clutter map-related thresholding system only (NO CFAR).
STIPL	Logical variable. Set STIPL = T if a plot of S/I vs. target doppler frequency is desired
RHOG	Set RHOG = 1 in all runs. This implies a unit ground clutter correlation from range cell to range cell.
THMDL	See CFAR.
XMIN	Minimum frequency (x-coordinate) for plotting routine
XMAX	Maximum frequency (x-coordinate) for plotting routine
YMIN	Minimum y-coordinate for plotting routine.
YMAX	Maximum y-coordinate for plotting routine.

The operator inputs to CFAR and CFARP are all done via namelist statements. Example inputs for CFAR and CFARP are shown below:

CFAR

\$INP1, NPTS = 64, WMDL = 1, MTI = T, MTIMDL = 1, THMDL = 1,  
STNDB = 0., STGCDB = -35., A = 2., IWGT = T, SIGSG = 18,  
SIGSW = 18., GM = T, NM = T, WM = F, NBAR = 6, ATAY = 1.6864\$



#### CFARP

\$INP1, RHOG = 1, NRCELLS = 32, NCLMAP = 6, NPTS = 64,  
MTIMDL = 1, ITATTN = F, IATTN = F, IFRESP = F, MORCFAR = T,  
MTI = T, IWGT = T, STIPL = F, THMDL = 1, PRFA = F, IDEALCF = T,  
PRED = T\$  
\$INP2, IPROB = T, XMIN = 0., XMAX = 5000.\$

Finally, in specifying varying degrees of Taylor weighting, the parameters ATAY and NBAR defined in Table 4-1 are required. These parameters are given below in Table 4-3 for different sidelobe levels.

Table 4-3. Taylor Weight Parameters

Sidelobe Level	A	NBAR
-20	.9527	3
-30	1.3197	4
-40	1.6864	6
-50	2.055	9
-60	2.42	12
-65	2.605	14
-70	2.785	17
-75	2.97	19
-80	3.15	21
-85	3.335	22
-90	3.52	25

#### 4.3.3 Simulation Outputs

##### CFAR Outputs

The CFAR program produces very little printed output since its main purpose is to generate and store statistics that are needed by the CFARP program. Those outputs printed by CFAR are:

- All Namelist Inputs
- SIGCSQ ARRAY (normalized<sup>(1)</sup> ground clutter cross section each FFT filter)
- SIGNSQ ARRAY (normalized noise power each FFT filter)
- SIGWSQ ARRAY (normalized rain cross section each FFT filter)
- MUT, MUCL, KCL, MUN, MUCLF<sup>(2)</sup>, MU1, MUW1= MUW2

where MUT = Target cross section ( $M^2$ ), MUCL = total ground clutter cross section, KCL = cross section of point clutter (always 0), MUN = mean noise power, MUCLF = cross section of fluctuating ground clutter component (0) and the remaining variables are defined in Table 4-1.

#### CFARP Outputs

The main CFARP output is detection probability vs. target doppler frequency for given signal-to-interference inputs and a given clutter map/CFAR signal processor configuration. Also available for output are detection probabilities for fixed threshold (known interference) systems. Such a system is one in which each FFT filter threshold varies from range bin to range bin but is constant in any given bin for all time. A system of this type thus defines a limit to the performance of any corresponding CFAR system and can be used as a reference for establishing a loss in S/N due to CFAR (CFAR loss).

Some of the secondary outputs of the program (which were included primarily as an aid in the initial debugging process) include:

- Clutter map false alarm constant
- CFAR false alarm constant
- A plot of the MTI response vs. frequency. The plots are normalized to the noise output of the zeroth FFT filter
- Plot of clutter map attenuation vs. frequency
- Plot of S/I vs. target doppler frequency

---

(1) Normalization is with respect to the appropriate cross sections at the signal processor input.

(2) The MUCLF variable is currently not used by the program. Since there is yet no point clutter model, MUCL represents the fluctuating component. Until a point plus fluctuating clutter model is implemented, a zero will be printed for MUCLF.

- Plot of ground clutter + noise attenuation vs. frequency
- Plot of false alarm probability at each FFT filter for clutter map related thresholding (see Section 4.2.3)

#### 4.4 Simulation Results

##### 4.4.1 Introduction

In this section, the results of selected computer simulation runs are presented. The objective here is to first compare the detection performance of several fixed threshold signal processors in a heavy ground clutter environment. Since these fixed threshold systems define a limit to the performance of any corresponding CFAR system, the comparisons can be used to define the basic components and parameters of a clutter map/CFAR signal processor that would be effective in comparable clutter conditions. The predicted performance of such a clutter map/CFAR system is then presented and compared with the performance of the corresponding fixed threshold system for various S/N and S/C conditions. The comparison is then extended to include the effect of weather clutter on detection performance.

##### 4.4.2 Performance Curves

Figure 4-4 shows a comparison between each of the fixed threshold systems:

- a. No MTI and no pre-FFT weighting
- b. No MTI, but 40 dB Taylor weights
- c. 3 pulse sliding window MTI and no pre-FFT weights
- d. 3 pulse sliding window MTI and 40 dB Taylor weights
- e. Narrow notch MTI\* and no pre-FFT weighting
- f. Narrow notch MTI and 40 dB Taylor weights
- g. 3 pulse sliding window MTI and no FFT.

A 64-point FFT was assumed along with a Swerling 1 target, S/N and S/C conditions of 10 dB and -35 dB, respectively, and ground clutter with a spectral width of 18 Hz. Also, the curves in this figure reflect an average false alarm probability of  $10^{-6}$ . It is seen that for the given interference conditions a low detection

---

\* "Narrow-notch" MTI is an MTI in which the average of the L pulses required for an L-point FFT is subtracted from each pulse before entering the FFT. For non-scanning antenna and completely non-moving clutter, this type of MTI is optimum.

probability, reaching a maximum of about 46%, is maintained over all target doppler frequencies in the case of no MTI nor pre-FFT weighting. Simply adding 40 dB Taylor weights before the FFT increases the detection probability on the average by about 45 percentage points in the higher filters, and a detection probability of more than 80% is maintained in 60% of the filters (target dopplers between 1000 and 4000 Hz). On the other hand, the detection probability is still less than 70% for target dopplers less than 600 Hz (i. e., for targets appearing within the first 8 FFT filters).

The highest detection probabilities shown are for the case of a 3 pulse MTI and no pre FFT weighting. In this case, detection probabilities larger than 97.5% are achieved 72% of the time, and less than 70% detection occurs only about 8% of the time for target dopplers less than about 200 Hz (1st 3 filters).

Following the MTI with 40 dB Taylor weighting degrades detection performance slightly in the higher filters, but improves performance significantly in the lower filters. Better than 80% detection occurs beyond the 2nd FFT filter and better than 90% detection beyond the 3rd. The degradation in detection performance in the higher filters is less than 1% and is expected since a reduction in S/N accompanies pre FFT weighting.

Also shown in this figure is the significant improvement in detection of the MTI/doppler processors over the simple MTI systems in strong clutter environment.

In order to interpret the results of Figure 4-4 in terms of S/N in dB, we resort to the theoretical developments in Section 4.2. From Equations (4-22) and (4-41), the probabilities of false alarm and detection for fixed threshold systems (Swerling I target) are given respectively by -

$$\begin{aligned} \text{PFA} &= e^{-y_b} \\ \text{PD} &= e^{-y_b/(1 + \bar{x}_F)} = (\text{PFA})^{\frac{1}{1 + \bar{x}_F}} \end{aligned} \quad (4-44)$$

where  $y_b$  is the threshold and  $\bar{x}_F$  is the signal-to-noise ratio required in the fixed threshold case. Here we restrict the conversation to the higher FFT filters where ground clutter effects are negligible<sup>(1)</sup>. For these filters, (4-44) leads to the expression -

---

(1) In general  $\bar{x}_F$  is the average S/L

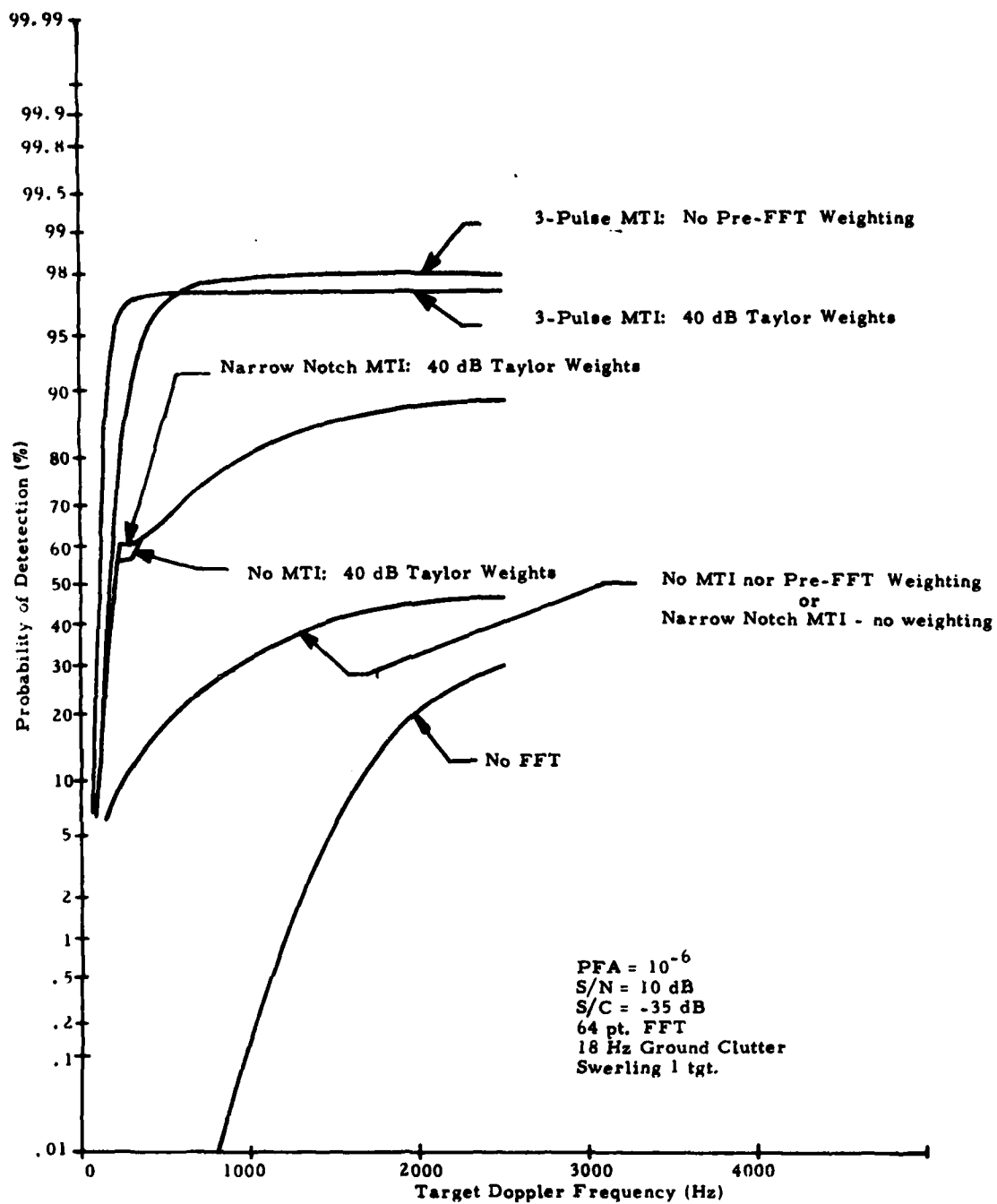


Figure 4-4. Probability of Detection: Ideal CFAR

$$\bar{x}_F = \frac{\text{Ln}(\text{PFA})}{\text{Ln}(\text{PD})} - 1 \quad (4-45)$$

for the signal-to-noise ratio required for given false alarm rates and detection probabilities.

Paralleling the above for CFAR systems, Equations (4-26) and (4-42) lead to the expression

$$\bar{x}_c = \frac{(\bar{\text{PFA}})^{-\frac{1}{N}} - 1}{(\bar{\text{PD}})^{-\frac{1}{N}} - 1} - 1 \quad (4-46)$$

$N$  # Range cells averaged

for the average S/N required in the higher filters for given average false alarm rates and detection probabilities. Again, it should be pointed out that these expressions are valid for Swerling I targets only and single pulse hits out of the square law magnitude unit following the FFT.

Equations (4-45) and (4-46) lead to the curves plotted in Figure 4-5 for a false alarm probability of  $10^{-6}$ . Applying the fixed threshold curve in Figure 4-5 to the curves in Figure 4-4 suggests that an increase of more than 14.5 dB in signal-to-noise is achieved in the highest FFT filters by adding the MTI and 40 dB weights to the no-MTI, no-weighting system. Also, the degradation in detection performance of less than 1% (for higher filters) when 40 dB weighting is added to the MTI system corresponds to a loss of about 1.2 dB in S/N.

Also indicated in Figure 4-4 are results for a narrow notch MTI, that is, an MTI in which the average of the L-pulses required for an L-point FFT is subtracted from each pulse before entering the FFT. It is not surprising that there is essentially no improvement over the no-MTI case, as the notch is just too narrow to eliminate all but a D. C. component of ground clutter. It also does an exceptionally good job of eliminating the targets in the zero doppler bin. Figure 4-6 clearly shows this to be true. In this figure, the relative responses of the 3-pulse and narrow notch MTI are compared. Also shown is the expected roll-off in frequency of 18 Hz ground clutter.

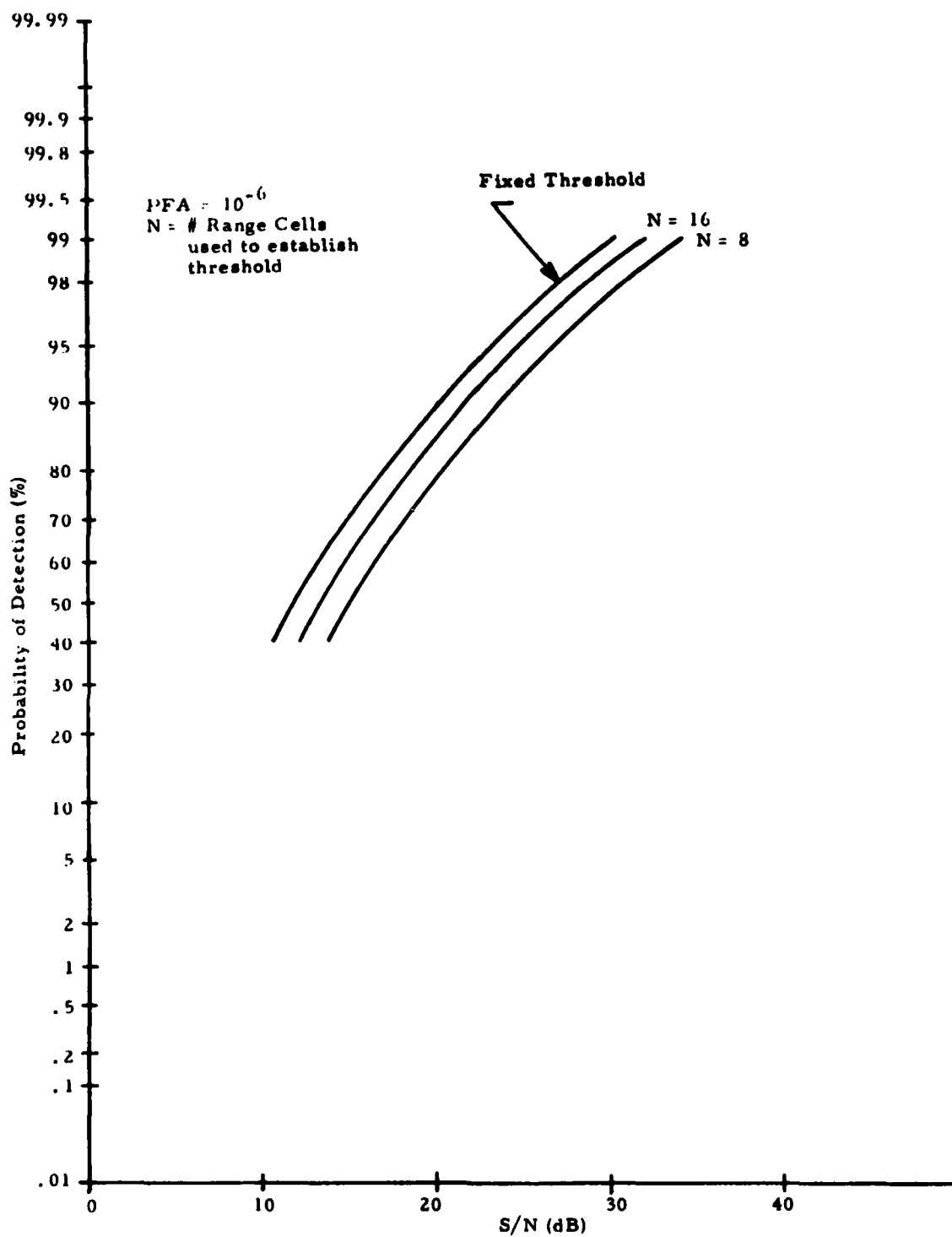


Figure 4-5. S/N Requirements for Detection

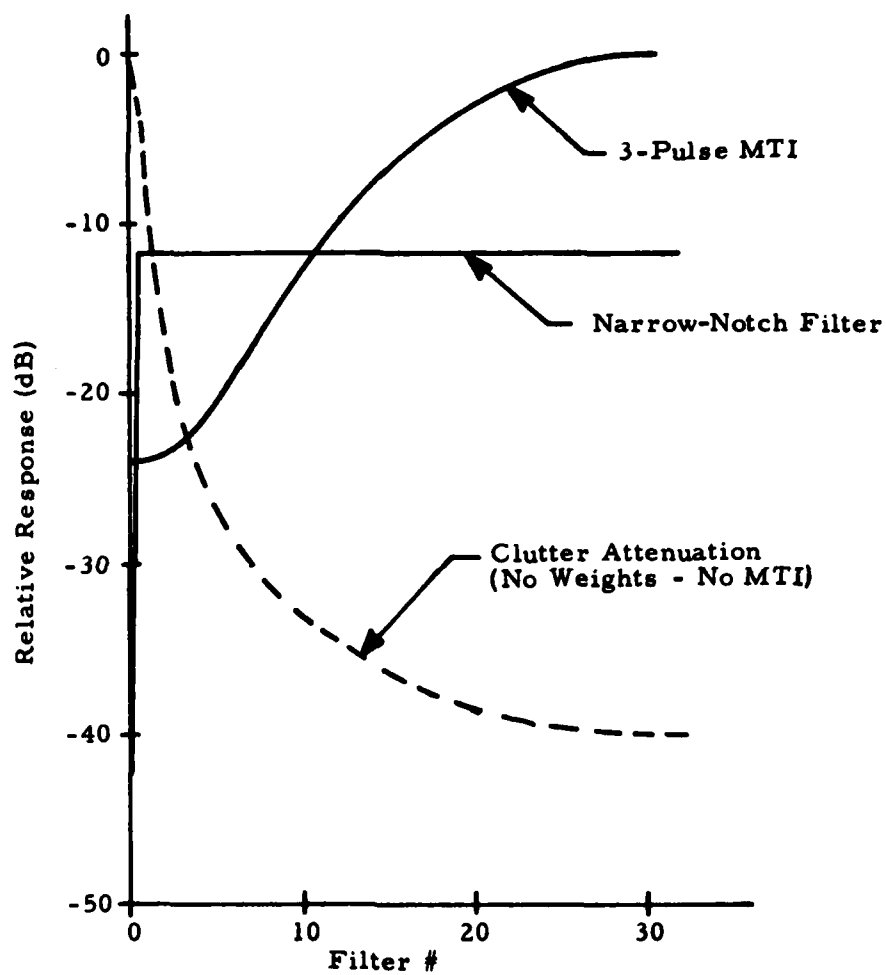


Figure 4-6. Relative Response Of 3-Pulse and Narrow-Notch Filters



Figure 4-7 shows the effect of changing slightly the degree of pre-FFT weighting for the fixed threshold, 3-pulse MTI system. By varying the 40 dB Taylor weights by  $\pm 10$  dB, the signal-to-noise requirements for the same level of detection in the higher filters vary by about  $\pm .5$  dB. That is, the CFAR loss is  $\pm .5$  dB. Also plotted in this figure is the detection curve for a 16 pt. FFT based on a single pulse out of the magnitude unit. In order to present a fair comparison between the 16 and 64 pt. FFT, non-coherent integration of 4 signal plus noise variates from the 16 pt. FFT should be performed. When this is done, the resulting detection curve for the Swerling I target is raised about 5.5 percentage points above the single pulse result (see Reference [11], page 238) for the higher target dopplers, but is still somewhat below the detection curve for the 64 pt. transform. If, however, frequency diversity were used to produce rapid pulse-to-pulse fluctuations out of the magnitude unit, Swerling II curves from [11] indicate that only 2 pulses are necessary to raise the detection curve for the 16-point transform above that for the 64 pt. transform. This would then be an alternative to using the 64-point FFT in heavy clutter environments.

Performance curves for a clutter map/CFAR system consisting of a 3-pulse MTI, 40 dB Taylor weights, and a 64-point FFT are shown in Figure 4-8 along with the corresponding fixed threshold case. Also shown are performance curves for simple MTI (No FFT) systems. Here the same heavy clutter environment as in the previous figures is assumed. For doppler frequencies  $> 50.0$  Hz, the CFAR loss in dB can be read off the curves in Figure 4-5. As the number of range cells averaged, each side of the target cell, decreases from 32 to 8, the CFAR loss increases but still remains less than 1 dB. Similar performance curves for various S/N and S/C ratios are given in Figure 4-9.

Finally, Figures 4-10 and 4-11 show clutter map/CFAR performance for an MTI, 40 dB weighting, 64 pt. FFT system operating in heavy ground clutter and rain. The results are given for various rainfall rates and the extreme cases of storms with center frequencies of 0 and 945 Hz.

#### 4.4.3 Remarks and Conclusions

- a. In heavy clutter environments, a combination of MTI and appropriate pre-FFT weighting will improve the detection performance over a system which excludes the MTI

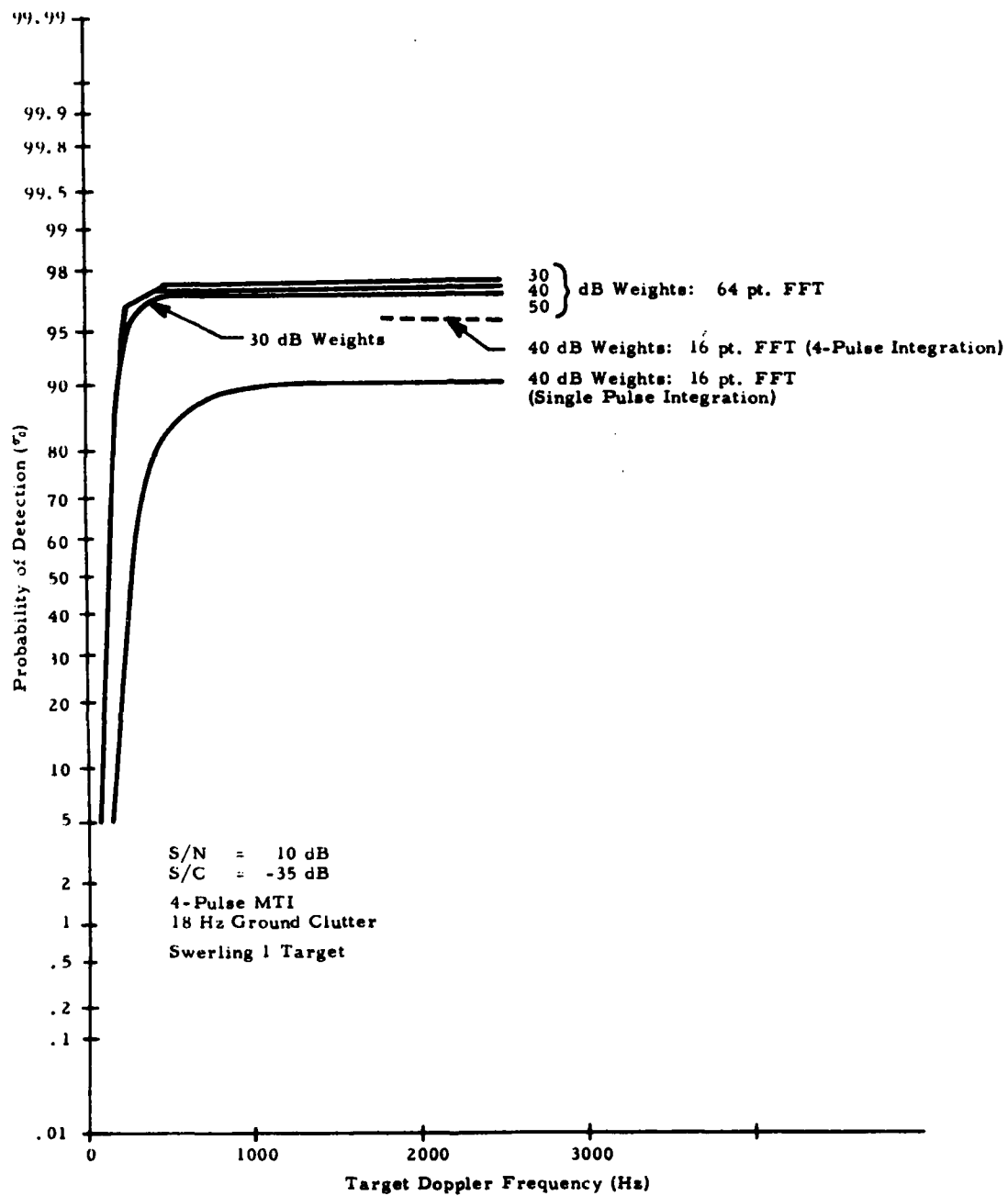


Figure 4-7. Probability of Detection: Fixed Threshold

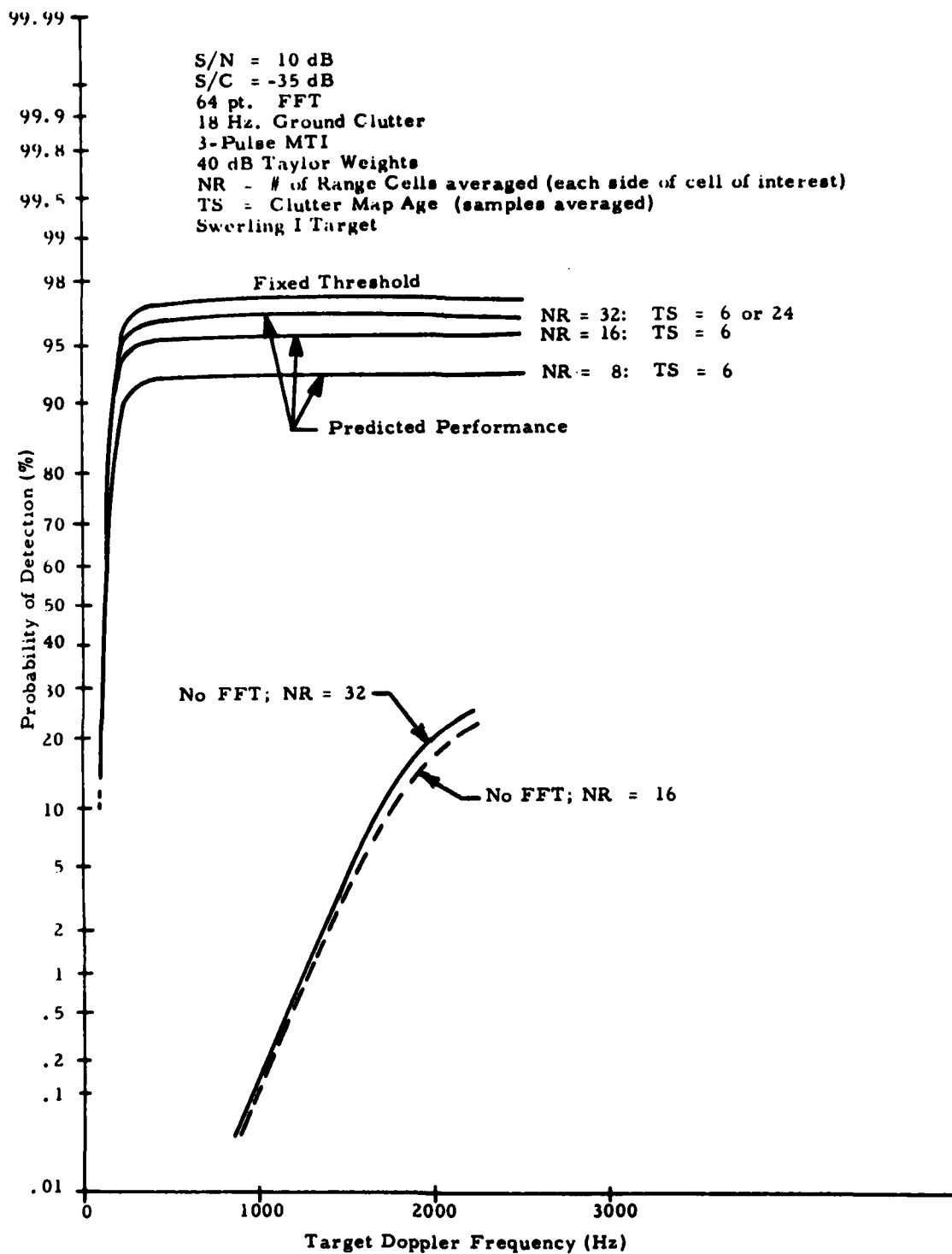


Figure 4-8. Clutter Map/CFAR Performance Curves  
 (Noise & Ground Clutter Environment)

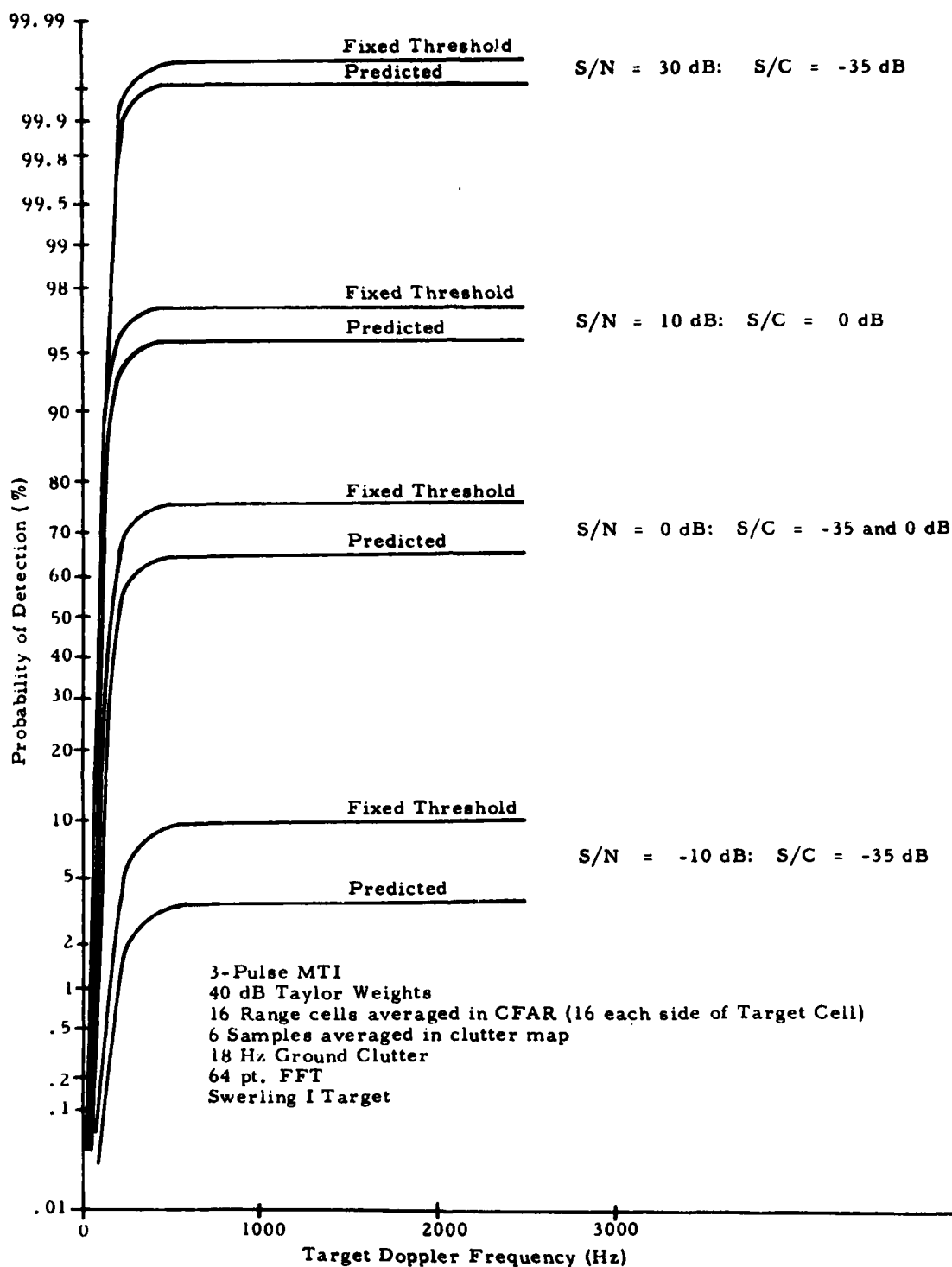


Figure 4-9. Clutter Map/CFAR Performance Curves  
(Noise & Ground Clutter Environment)

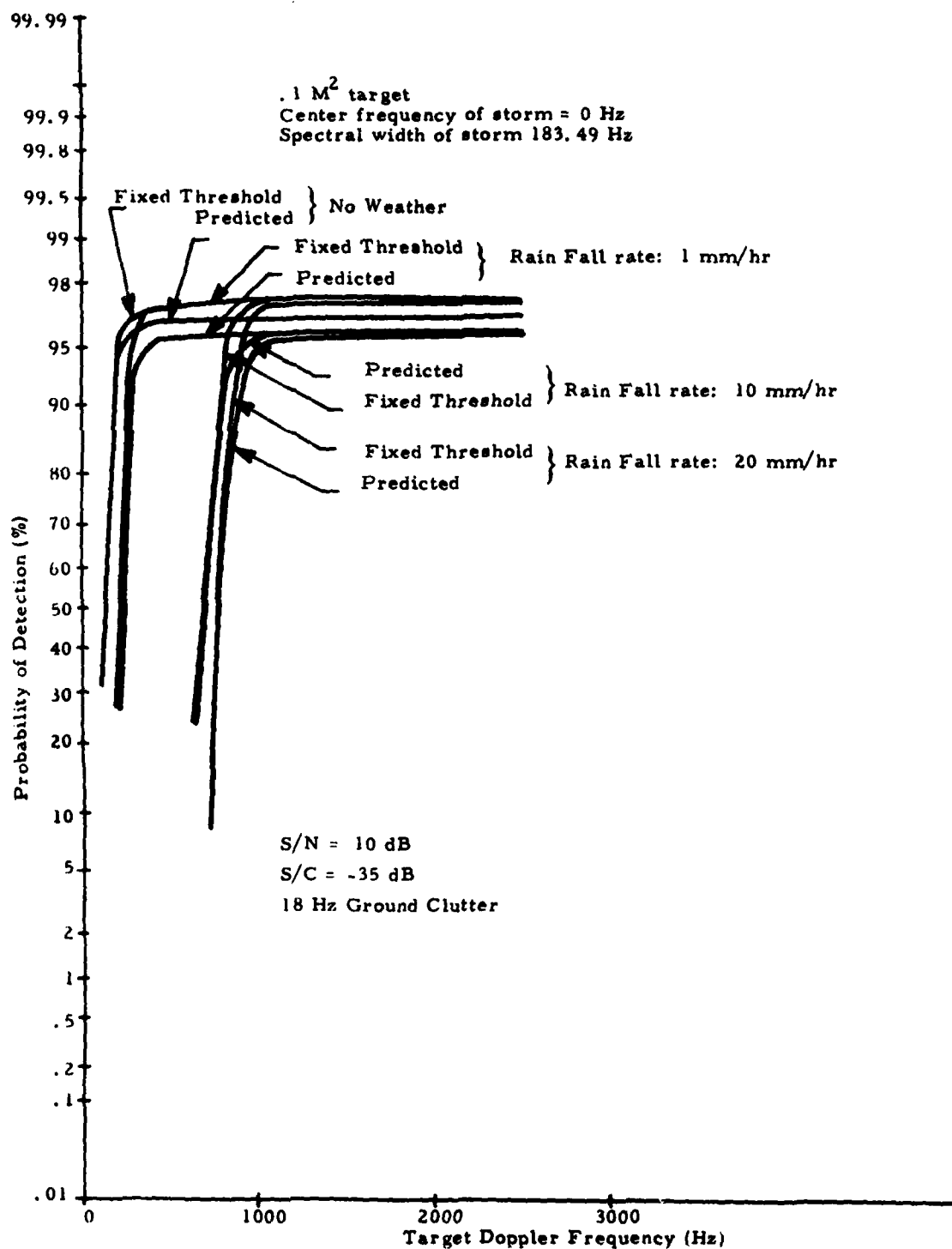


Figure 4-10. Clutter Map/CFAR Performance Curves (Noise, Ground Clutter, Weather Environment)

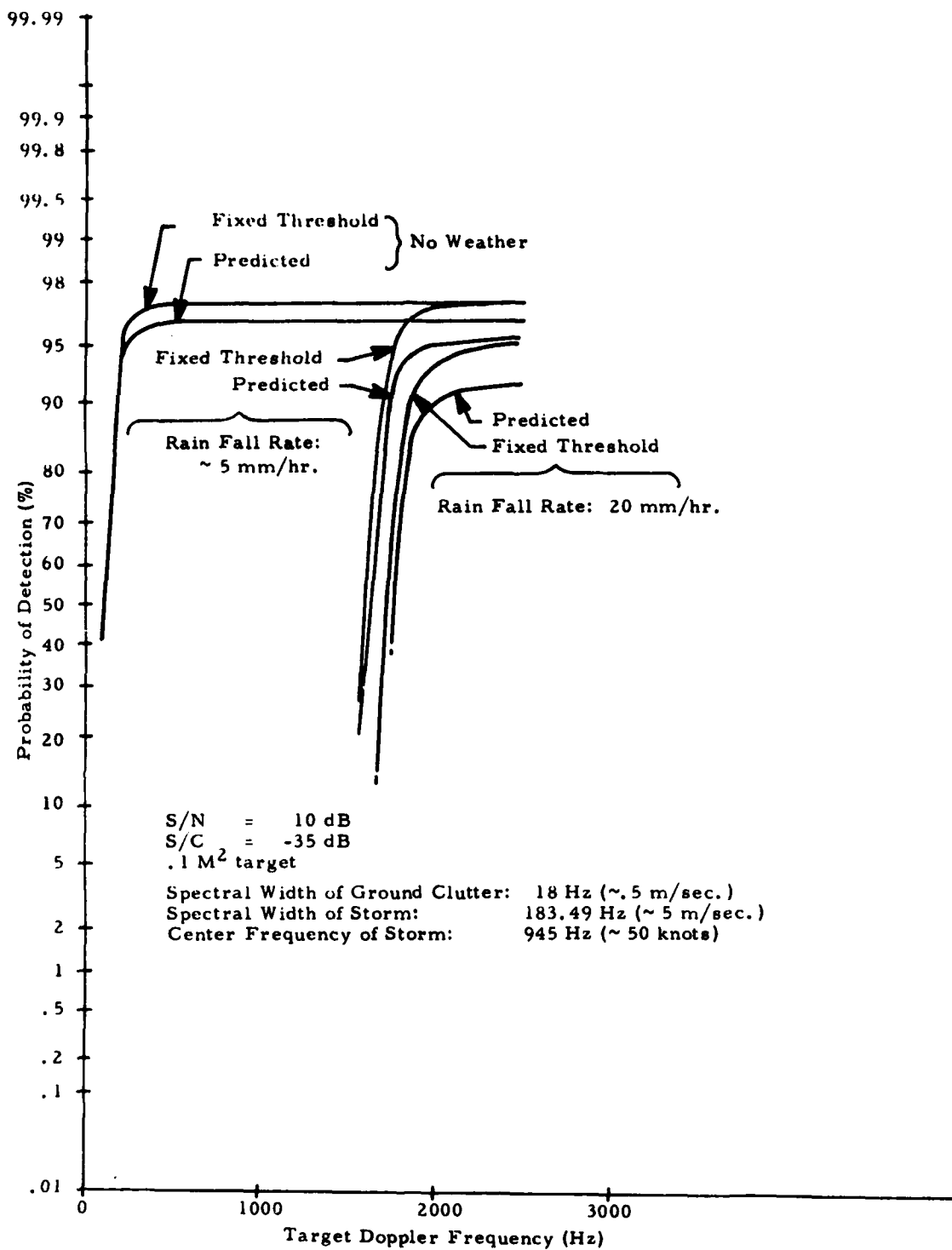


Figure 4-11. Clutter Map/CFAR Performance Curves  
(Noise, Ground Clutter, Weather Environment)

- b. Low-velocity targets or targets near a blind speed are lost or severely degraded by the MTI. Several things can be done to improve this situation.
  - Implement a range-gated MTI, such that clutter is sensed in each range cell and the MTI activated only if the clutter amplitude is significant.
  - Implement a staggered PRF MTI to minimize blind speed effects.
  - We have considered two MTI configurations here: the three-pulse MTI and the narrow notch MTI. The three-pulse MTI is much better than the other in clutter rejection but worse in the rejection of low velocity targets. Neither is really optimum for the assumed target and clutter environment. Further study is needed to determine the design of the optimum MTI for this processor.
- c. The curves presented do not include the effect of strong point clutter on target detection. Thought should be given toward refining the simulation to include this D. C. clutter component.
- d. The simulation effectively handles Swerling I targets based on one hit only. Non-coherent integration and the capability of handling other Swerling targets should be included as future refinements to the simulation.
- e. At present, the rain model is reasonable for zero center frequency cases and cases where the center frequency of the rain storm is large compared to the spectral width of the rain. Some thought should be given toward refining the model to include intermediate cases.

## SECTION 5. HARDWARE CONFIGURATIONS AND TRADEOFFS

### 5.1 Introduction

Having, in the preceding section, determined the requirements on the FTS processor, we are left with the task of translating it into a hardware design. There is a variety of choices of processor architecture available. This section discusses the choices, their hardware requirements (in terms of components), software requirements, and degree of development necessary to fit them to the EAR needs. All of the approaches discussed here can meet the system requirements; the choice is then to be made on the basis of relative cost and design flexibility.

### 5.2 Input Interface

The A/D converters of the signal processor convert the bipolar video radar returns (of the Inphase and Quadrature channels) to a suitable binary format for digital processing. These converters (existing equipment) are Computer Lab Model HS-905 9-bit converters. They will be operated in the EAR at the rated (sample) 5 MHz rate with a dynamic range of  $\pm 1024$  MV. The output format is an offset binary code where

1	1111	1111	denotes + 1024 MV
1	0000	0000	denotes 0000 MV
0	0000	0000	denotes - 1024 MV

The A/D converter outputs and the input code commands are transferred to and from the Digital Moving Target Indicator (DMTI)\* logic through a digital buffer unit consisting of digital differential line drivers and receivers. The output from the A/D's can bypass the DMTI (upon command) and input directly to the FTS processor.

---

\* This refers to the existing DMTI, rather than to any new sliding window DMTI which might be implemented as part of the FTS processor.



### 5.3 Arithmetic Precision and Dynamic Range Considerations

The existing 9-bit (8 bits plus sign) A/D converters define the dynamic range capability of the processor; allowing 2 or 3 bits to define the rms noise level leaves 5 or 6 bits for target amplitude variation, implying that the maximum signal-to-noise ratio that the processor can handle linearly is 30 to 35 dB. The ratio between largest linear signal and smallest detectable signal can be greater than this, however, since the coherent integration due to the FFT will allow signals weaker than the input noise level to be detected. For example a 64-point FFT produces 18 dB of S/N gain; for a 13 dB output S/N (consistent with reasonable probability of detection and false alarm rate), the input S/N is -5 dB.

We can expect strong point clutter to sometimes exceed the linear range of the processor. It is important that this overload be handled in the right way. The limiting should be defined in the IF circuitry of the radar, before the I and Q phase detectors. If the A/D converters themselves are allowed to be overdriven, undesirable non-linear effects will take place (i.e., clutter harmonics being generated in higher frequency doppler bands) which can seriously degrade system performance. With limiting handled correctly at IF, the clutter map circuitry in the FTS processor can eliminate the clutter point from the output.

Although the processor input is 9 bits, the number of bits within the processor grows by integration. This fact plus practical considerations relating to the bit specifications of available integrated circuits dictate that internal arithmetic be carried out to 16 bits, although FFT coefficients and sidelobe weighting coefficients need not be carried higher than 12 bits.

### 5.4 Input Weighting

The weighting function is multiplied by the complex input data stream before it is stored (as a block of N samples per range gate) in the input buffer. The input data is generated as an offset binary code. For efficient multiplication it must be converted to a 2's complement representation. The weighting coefficients are stored as 12-bit numbers; every 200 ns two

10 x 12 bit multiplications are performed with the resulting output rounded to a 10-bit 2's complement number before storage in the buffer.

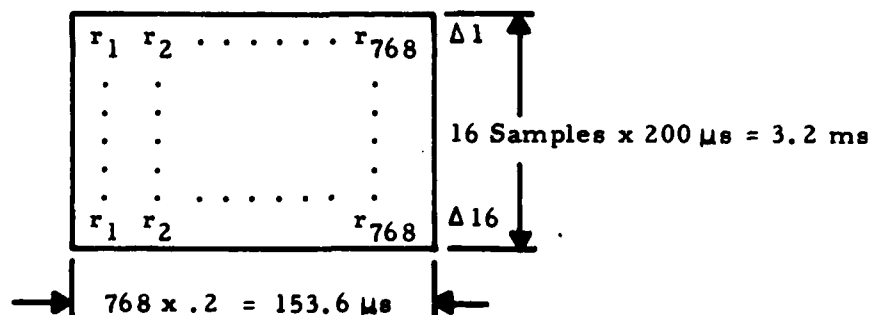
### 5.5 Input Buffer

The input buffer must be sized to handle the maximum number of range bins times the number of points in the FFT. Flexibility requirements dictate that the number of points be variable. Then we will need to vary the number of range cells that can be processed as well if the buffer size is to be constant. Each range cell generates a 20-bit word (10 bits each for I and Q channels). Then for example if we consider 768 range gates (bins) the maximum, and the minimum transform size as 16 points, then

768 range gates	x	16 points	} = 12,288 words x 20 bits
384 range gates	x	32 points	
192 range gates	x	64 points	
96 range gates	x	128 points	
48 range gates	x	256 points	
24 range gates	x	512 points	
12 range gates	x	1024 points	

We choose the closest binary value, 16,384 words x 20 bits or 327,680 (this excess will provide for storage of intermediate values when performing the Fast Fourier Transform). The binary approach to varying the range gates also keeps the control mechanism of the input buffer simple when writing into or reading out of it.

Since the PRF is 5 kHz, the time between samples is 200  $\mu$ sec. For the 16 point transform the memory map appears as, (with the time to collect the data)



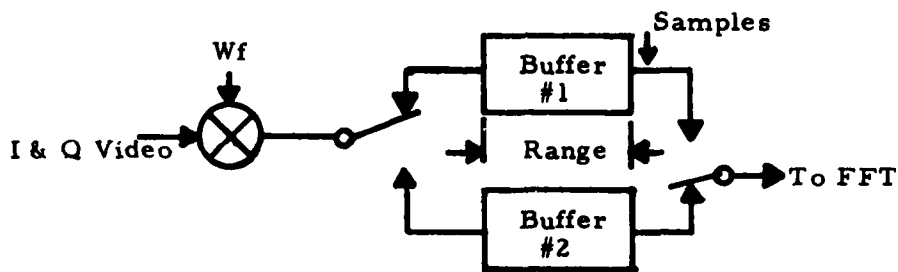
Although the buffer size is constant, the time to fill up a buffer is related to the number of points in the transform as follows:

768 range gates x	16 points	=	3.2 ms
384 range gates x	32 points	=	6.4 ms
192 range gates x	64 points	=	12.8 ms
96 range gates x	128 points	=	25.6 ms
48 range gates x	512 points	=	102.4 ms
24 range gates x	1024 points	=	204.8 ms

There are two possible techniques which can be used for the input buffer.

#### 5.5.1 Double Buffer Concept

After weighting, the digitized I & Q data is stored in one buffer while the second batch of data is fed into the FFT. This is depicted below and requires twice the memory size or  $2 \times 16,384 \text{ words} \times 20 \text{ bits} = 655,360 \text{ bits}$ .



The advantage of this technique is the simplicity of control; the disadvantage is that of requiring twice the memory size. Further detailed examination of the read/write speed required for this memory and its implementation is necessary. The Bipolar memory at this speed (5 MHz) comes in 1024 x 1 size and we can package 40 IC's on one card, or about 2K words x 20 bit on a single 8 x 10 inch PC board, for a total of 16 cards for the full 32K double buffer.

Another approach is to multiplex the data into a high-speed buffer (thus widening the word size) and transfer the wider word into a slower speed MOS memory. The output from the weighting network is that of a

10-bit number every 100 ns. If we distribute this by multiplexing into two sets of ten 10-bit registers at the 100 ns rate and then transferring one set into a slower MOS memory (100 bit word width), we can perhaps have a less costly solution than using all high-speed memory.

#### 5.5.2 Single Buffer With Corner-Turning Memory

This technique is a single buffer concept for storing the batch samples which are fed to the FFT. The basic corner-turning idea depends on the fact that, as data is read out of the memory, space is made available so that other data can be written in. At the end of a read/write sequence however the pattern of data (i. e., the address locations corresponding to a given range cell/sample point) has changed. The new pattern is however known to the sequence control program, and the whole process is repeated with a new set of addresses, leading to still a new pattern. And so on. Hopefully, in not too many repetitions the original pattern will repeat itself.

An investigation was made of the corner-turning pattern sequences involved for the projected memory. It was soon realized that the requirement for flexibility made the addressing problem involved in corner-turning hopelessly complex. Although for certain special cases simple pattern repetitions might be possible, it was certainly not true in general. A judgment was made that the complexity of the address generating circuitry would be sufficiently high in a flexible corner-turning memory so as to outweigh the saving in memory involved as compared to the double buffer.

### 5.6 Fast-Fourier Transform (FFT) Implementation

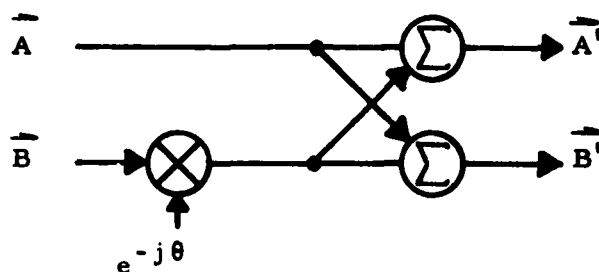
#### 5.6.1 General Structure

The well-known FFT algorithm efficiently carries out the Discrete Fourier Transform operation described by the following equation:

$$F(k) = \sum_{n=0}^{N-1} s(n) e^{-jnk} \left( \frac{2\pi}{N} \right) \quad (5-1)$$

where     $N$     =    number of points  
            $s(n)$  =    complex sample corresponding to  $n$ th point  
            $k$     =    filter number  
            $F(k)$  =    complex spectrum amplitude of  $k$ th filter

The FFT process by which the above equation is partitioned to minimize the necessary computations is abundantly described in the literature. Suffice it to say that the FFT requires about  $\frac{N}{2} \log_2 N$  complex arithmetic operations. The basic element of the FFT is the fundamental "Butterfly" operation which performs the necessary complex multiplications and summations, as shown below.



$$\vec{A} = I_A + j Q_A \quad (5-2)$$

$$\vec{B} = I_B + j Q_B$$

$$e^{-j\theta} = \cos \theta - j \sin \theta$$

$I$  = In-phase component

$Q$  = Quadrature component

$$\begin{aligned} \vec{A}' &= \left[ I_A + (I_B \cos \theta + Q_B \sin \theta) \right] + j \left[ Q_A + (Q_B \cos \theta - I_B \sin \theta) \right] \\ \vec{B}' &= \left[ I_A - (I_B \cos \theta + Q_B \sin \theta) \right] + j \left[ Q_A - (Q_B \cos \theta - I_B \sin \theta) \right] \end{aligned}$$

Figure 5-1. Butterfly Representation

Note from the above equations that the requirement is for 4 multiplication operations and 6 add/subtract operations. The four multiplication operations are:

$$I_B \times \cos \theta, Q_B \times \cos \theta, I_B \times \sin \theta, \text{ and } Q_B \times \sin \theta \quad (5-3)$$

The six add/subtract operations are:

$$\text{Partial Sum \#1} = I_B \cos \theta + Q_B \sin \theta \quad (5-4)$$

$$\text{Partial Sum \#2} = Q_B \cos \theta - I_B \sin \theta$$

$$I_A + \#1, I_A - \#1, Q_A + \#2, Q_A - \#2$$

Diagrammatically, the above mathematical process is represented as:

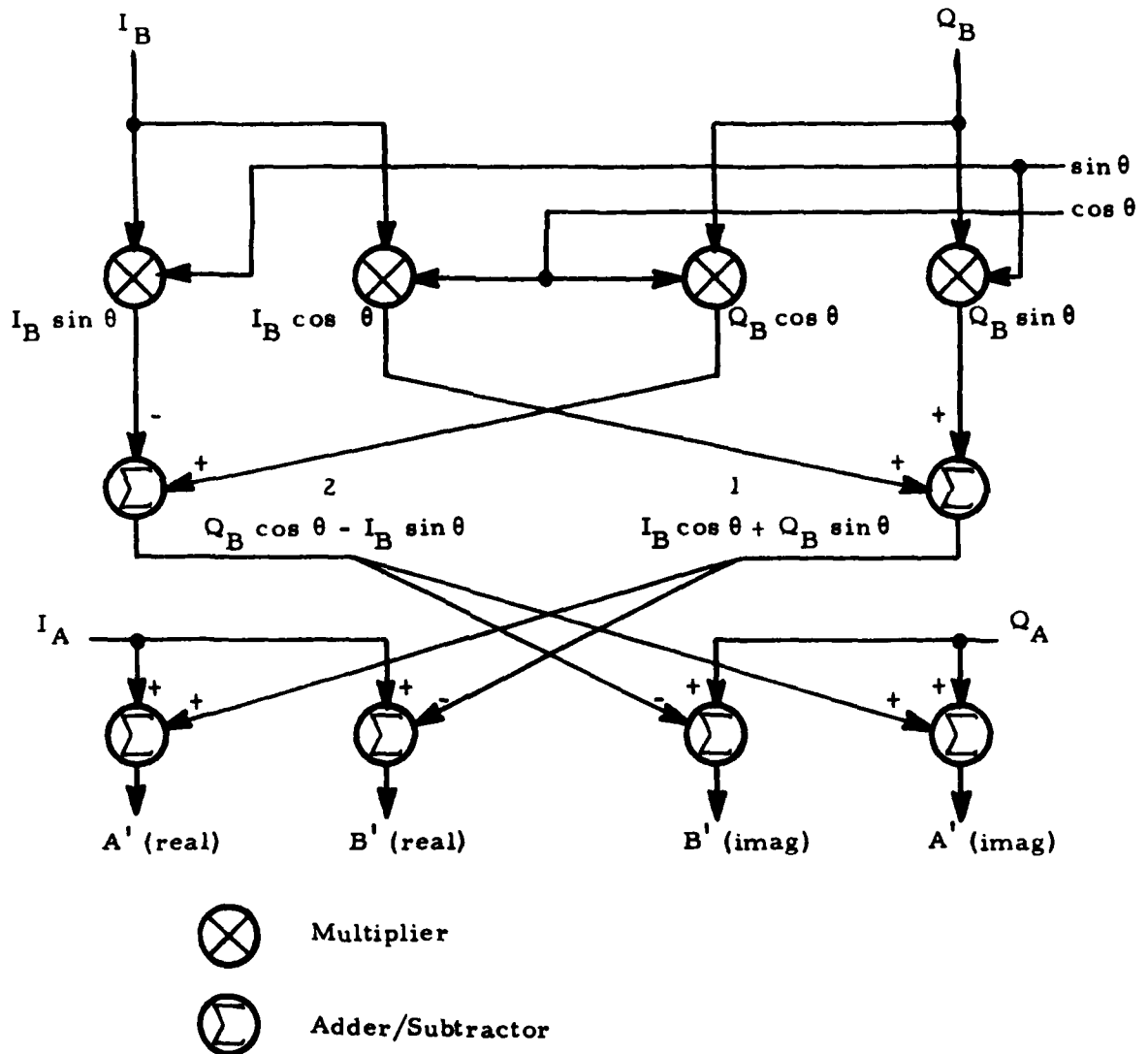
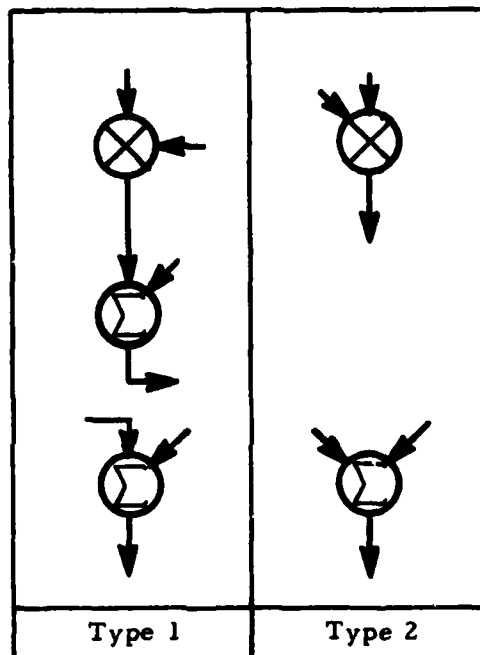


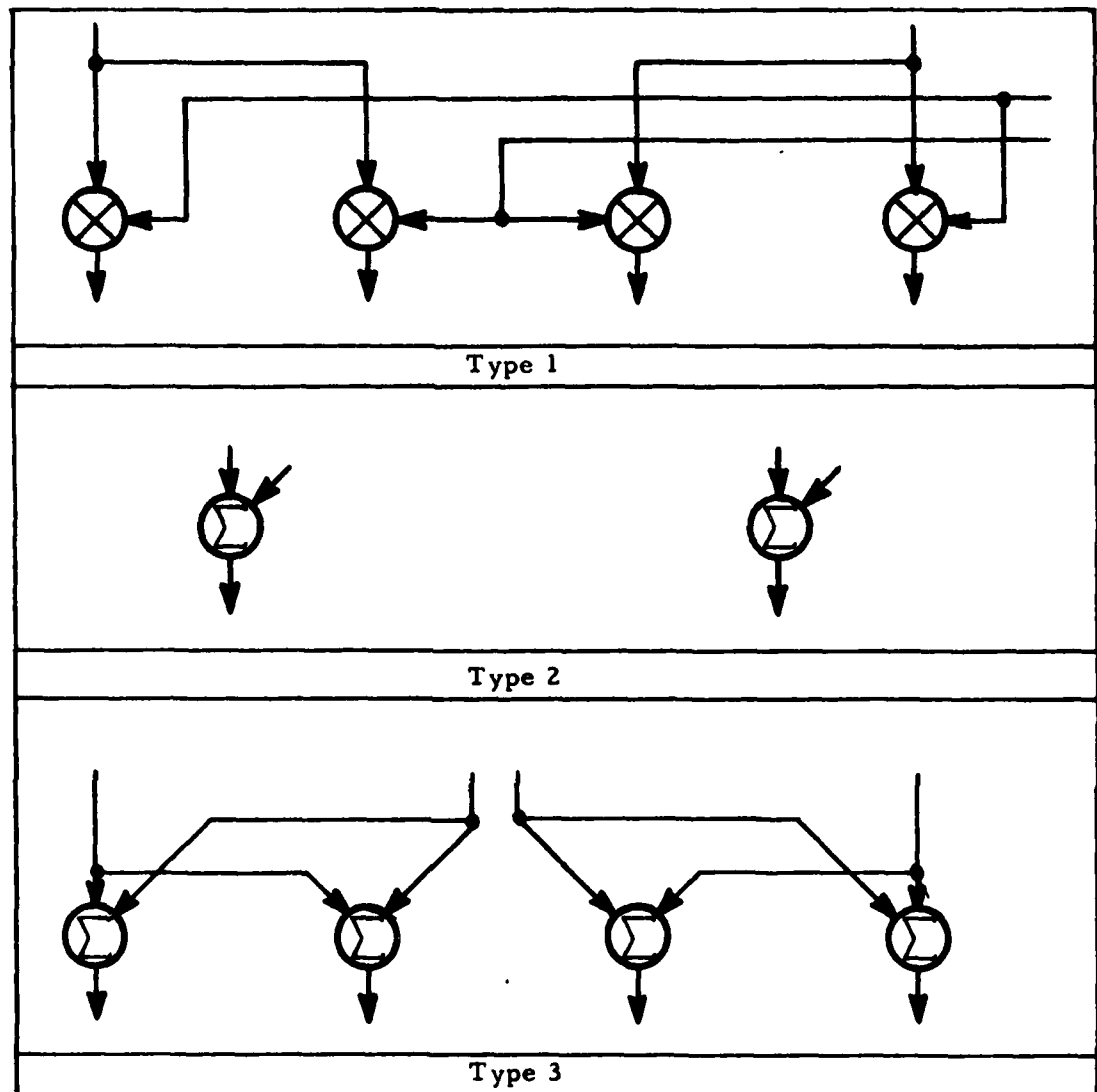
Figure 5-2. Physical Representation of Butterfly

Examining the above diagram, we see that the Butterfly structure can be divided, in a hardware organization sense, in two ways--vertical slice or horizontal slice.

Two kinds of vertical slice organization are diagrammed below. There can be either two types of cards--Type 1 and Type 2--or all Type 1 cards with one of the adders not used in half of the cards.



There can be three kinds of horizontal slice cards as shown below. Or we can get along with only Type 1 and Type 3, leaving out adders when a Type 2 is required.



### 5.6.2 Candidate FFT Approaches

Three general approaches to the FFT hardware design will be considered and compared. Two of them are basically existing Raytheon designs, although they must be modified and reprogrammed to fit the EAR requirements. These are called the Universal FFT (UFFT) and the General Purpose Signal Processor (GPSP). A third design is built around the use of common



processing elements tied together by a data distribution bus. Although potentially of high performance and flexibility, this approach requires more engineering development than the other two.

Still a fourth approach, rather specialized and somewhat out of the main stream of this study, involves the use of an existing Government-owned FFT. The characteristics of this processor and the problems involved in interfacing it with the EAR are discussed in Appendix B.

#### 5.6.2.1 The Universal FFT

The UFFT design (Raytheon Equipment Division) is representative of the vertical slice approach, with two type of cards. The design is based on a 2-bit recirculation (i. e., 2-bit multiplier and 2-bit adder) to provide the final result. The clock rate is 8 MHz (125 ns) with 8 passes through the multiplier/adder (8 x 125 ns) to provide a complete butterfly in 1  $\mu$ sec. A recent modification to this design has enabled a clock rate of 13.3 MHz (73 ns), speeding the butterfly up to 600 ns. The design as currently implemented includes the scratch pad RAM (for immediate storage of the two-bit partial results) and the control cards for a total of 12 cards (8" x 6").

#### 5.6.2.2 The General Purpose Signal Processor

Another approach to the FFT implementation is the GPSP design (Raytheon Missile Systems Division). This implementation provides a butterfly in 200 ns, using a pipeline construction. This architecture, shown in Figure 5-3, is a modification of the horizontal slice approach. This approach requires a minimum of 50 cards.

#### 5.6.2.3 The Common Element Approach

Another approach is to develop a common processing element (a micro-processor) which can do all of the necessary FFT operations for a certain number of range cells. As more range cells are needed, more common elements are added to the bus-interconnected system. This approach has a high degree of modularity and flexibility. It is diagrammed in Figure 5-4.

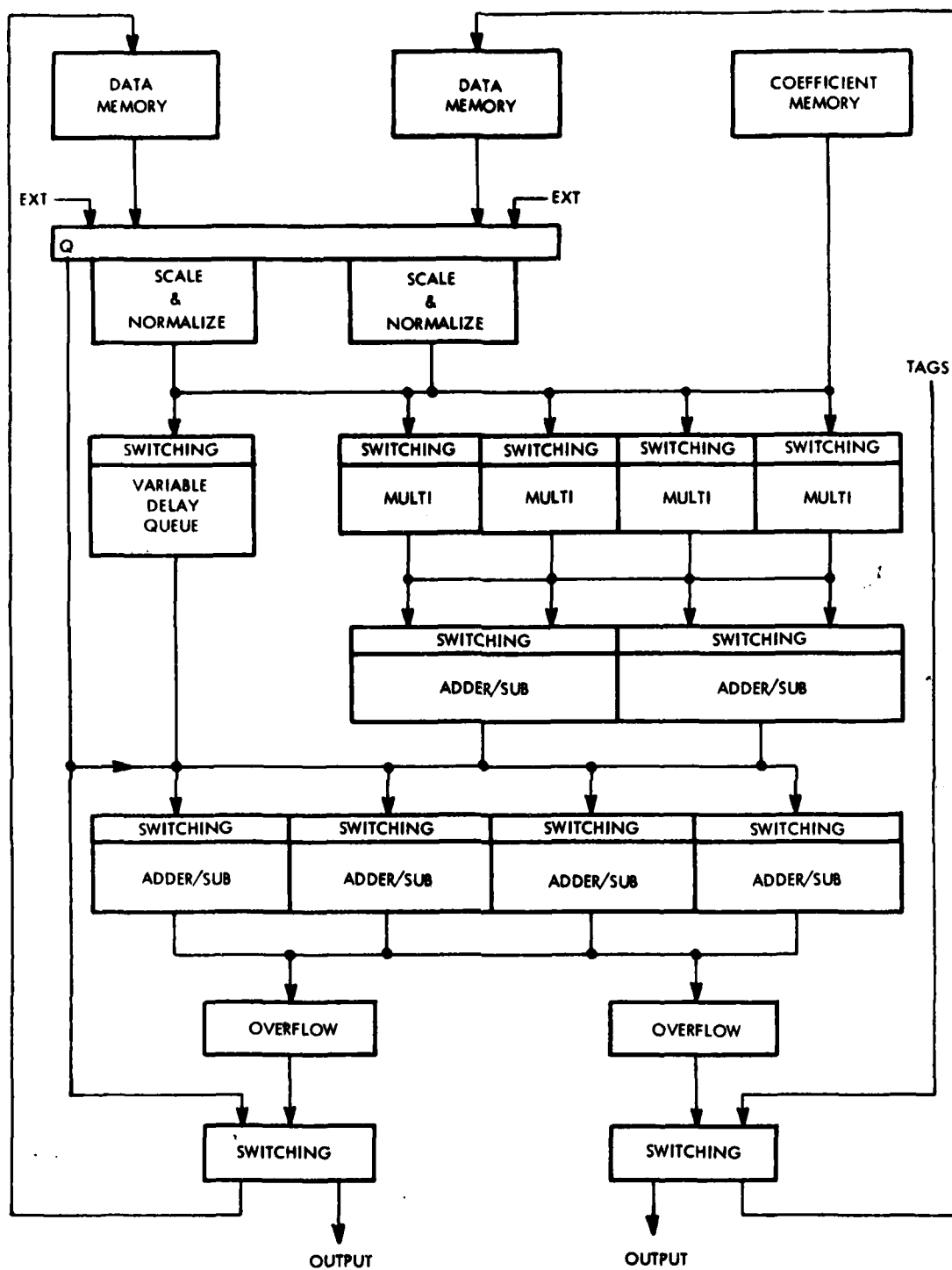


Figure 5-3. Programmable (Pipeline) Signal Processor

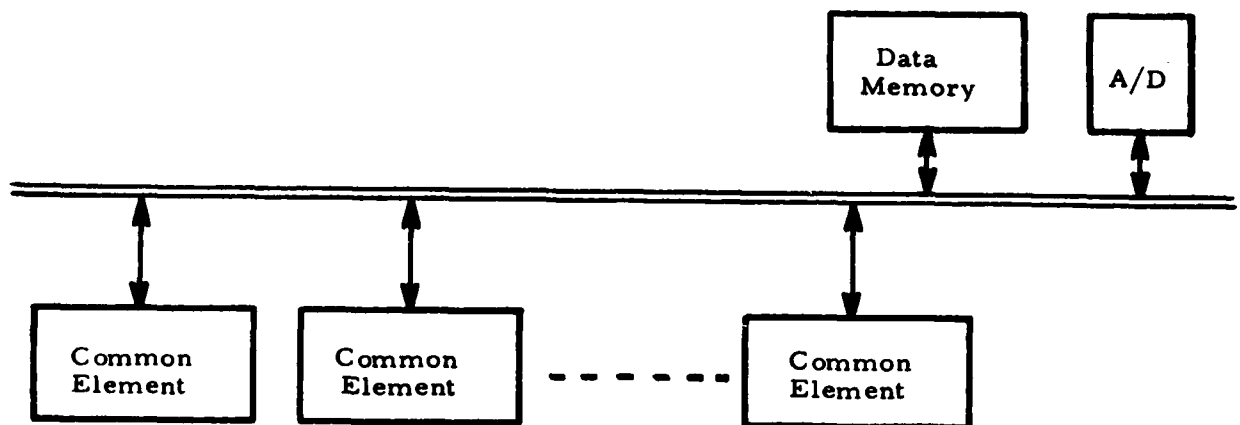


Figure 5-4. Distributed Signal Processor

The rationale for this scheme derives from the fact that in the FFT the I/O time is small compared to computation time. The common element approach also has this behavior. The processing time per range gate versus the number of points in the transform is given by the equation,

$$\text{processing time} = \frac{\text{number of points } (n)}{2} \times \log_2 n \times \text{time per Butterfly } (t_{BF})$$

The I/O time as a function of the number of points (n) is given by the equation,

$$\text{I/O time} = n \times 2 \times \text{time for I/O cycle } (t_{I/O}) \quad (5-5)$$

A third quantity of interest is the number of common elements necessary to perform the FFT process (as related to the Butterfly time) as a function of the number of points, and is found from the following:

$$\text{number of range gates/element} = \frac{\text{pulse repetition interval (time)} \times n}{\text{processing time for a single element}} \quad (5-6)$$

$$\text{number of elements} = \frac{\text{number of range gates in system}}{\text{number of range gates/element}} \quad (5-7)$$

A plot of processing time versus the number of points in the transform with  $t_{BF}$  as a parameter is shown in Figure 5-5. A plot of I/O time versus the transform size is shown in Figure 5-6. The two plots used together can show the ratio of I/O time to processing time for the FFT. For a 4  $\mu\text{sec}$  Butterfly time and a 60 ns cycle time of the memory:

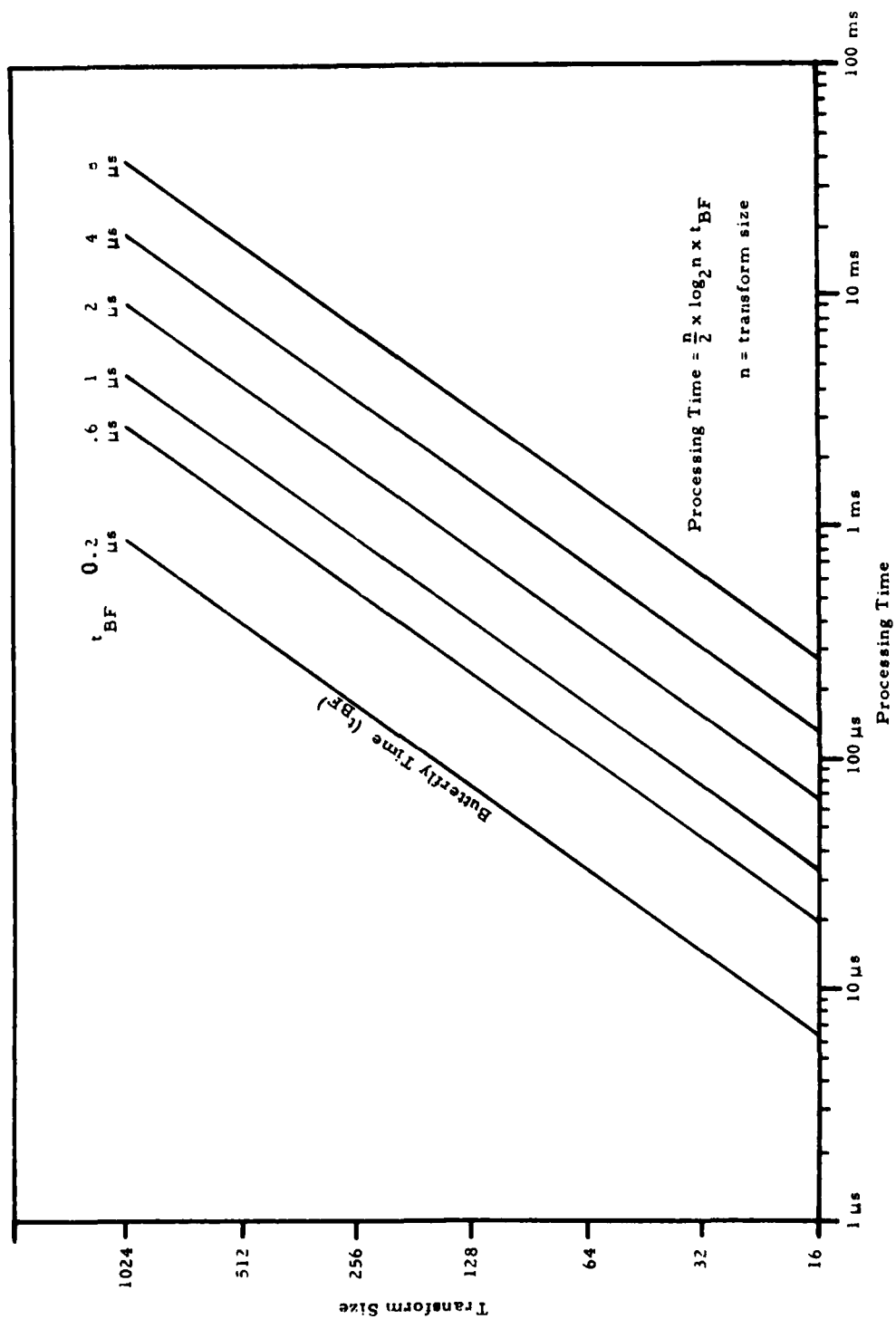


Figure 5-5. Processing Time Versus Transform Size

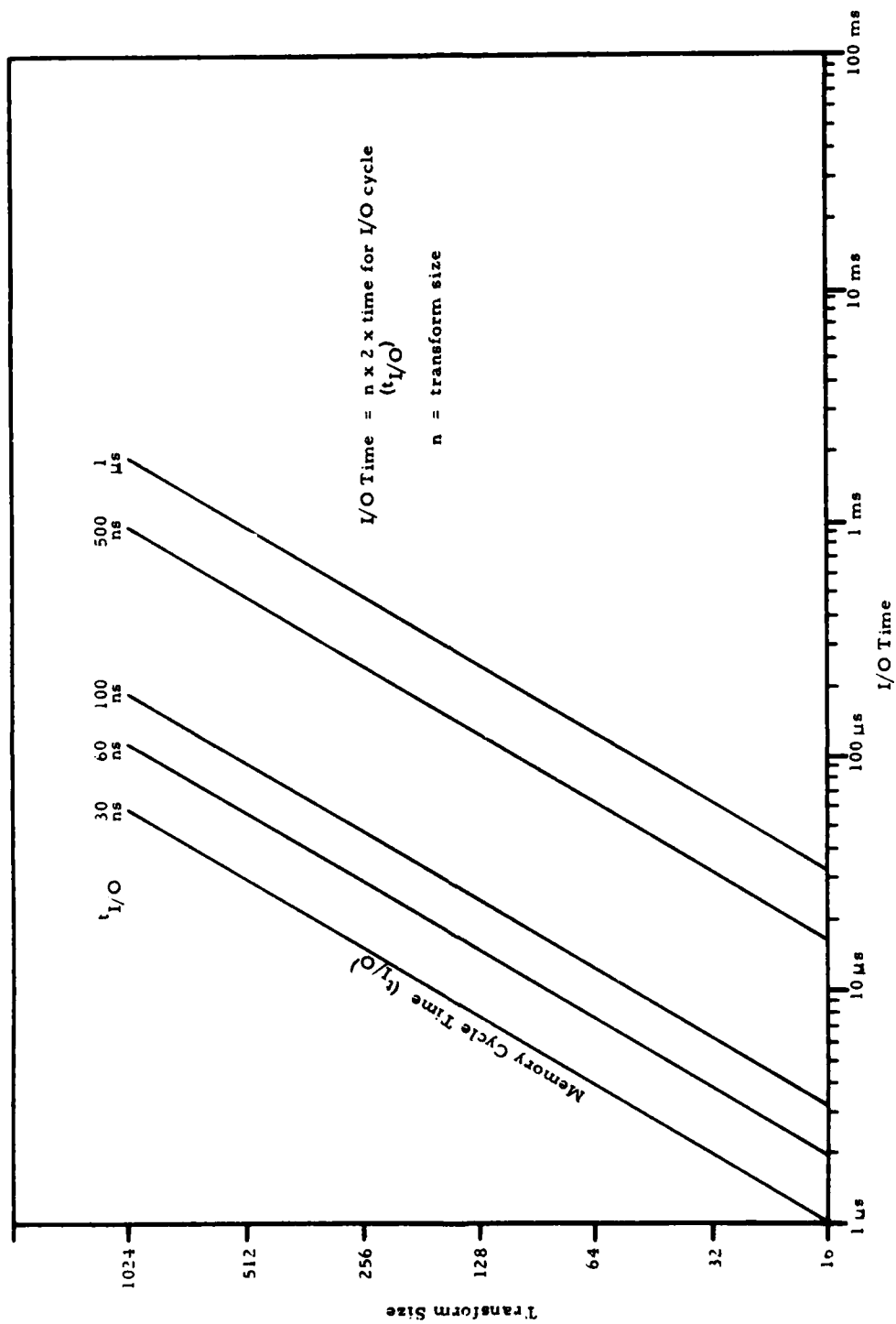


Figure 5-6. I/O Time Versus Transform Size

No. of pts	I/O Time	Processing Time	% = $\frac{\text{I/O Time}}{\text{Processing Time}}$
16	1.92 $\mu$ s	128 $\mu$ s	1.5%
32	3.84 $\mu$ s	320 $\mu$ s	1.2%
64	7.64 $\mu$ s	768 $\mu$ s	1.0%
128	15.3 $\mu$ s	1792 $\mu$ s	0.85%
256	30.72 $\mu$ s	4096 $\mu$ s	0.75%
512	61.44 $\mu$ s	9216 $\mu$ s	0.67%
1024	122.88 $\mu$ s	20480 $\mu$ s	0.6%

Using the common element approach to perform the FFT process then for 768 range gates, a PRF of 5 kHz and a 16-point transform then:

$$\text{number of range gates/element} = \frac{200 \mu\text{sec} \times 16}{128 \mu\text{sec}} = 25 \text{ ranges gates/element} \quad (5-8)$$

$$\text{number of elements} = \frac{768 \text{ range gates}}{25 \text{ range gates/element}} = 30.72 = 31 \text{ elements} \quad (5-9)$$

Using high-speed microprocessor techniques, one element can occupy one card. This approach if it can be properly realized in hardware, can represent a significant saving in total hardware.

#### 5.7 Signal Processing Hardware Requirements

This section summarizes the hardware requirements as a function of (a) The processing approach (GPSP, Common Element, UFFT/CPE\*), (b) the number of range gates processed, and (c) the number of FFT points processed. The various functions which need to be performed by the EAR Processor are:

- |                             |                              |
|-----------------------------|------------------------------|
| (a) Input Buffering         | (e) Range Averaging CFAR     |
| (b) Weighting               | (f) Frequency Averaging CFAR |
| (c) FFT                     | (g) Thresholding             |
| (d) Magnitude Determination | (h) Clutter Mapping          |

\*The CPE, or Common Processing Element, approach is a microprocessor-based technique developed at the Equipment Division for handling CFAR-type functions. It has been used in conjunction with the UFFT.

The GPSP and CE (Common Element) approaches can do all of these except (a) and (h). The UFFT does (b), (c) and (d). The CPE does (e), (f) and (g). (a) and (h) are basically memory card requirements with some processing (of the CE or CPE type) also needed in (h).

The way in which the various functions are handled by each of these approaches is illustrated in the functional block diagrams of Figures 5-7, 5-8, and 5-9. The functions are not explicitly shown in the Common Element Diagram (Figure 5-9) since all elements are basically the same, and are reprogrammed as needed to fulfill the necessary functions.

For each approach, relationships exist between the number of cards needed for each function and such system parameters as number of FFT points and number of range cells. These relationships are set down below; they are derived from known and estimated characteristics of these processors. Memory and clutter map requirements are common to all approaches and are discussed separately later.

#### Symbols

N	=	number of FFT points processed
R	=	number of range cells
I	=	number of operations/second
T <sub>o</sub>	=	scan time
B	=	number of FFT butterflies
C	=	number of cards (40-60 IC's each)
T <sub>p</sub>	=	processing time = $2 \times 10^{-4} N$ seconds for a 5 kHz PRF
INT	=	stands for "integral part of"

### 5.7.1 The Common Element Approach

#### 5.7.1.1 Weighting

Requires operations as follows: (per range gate)

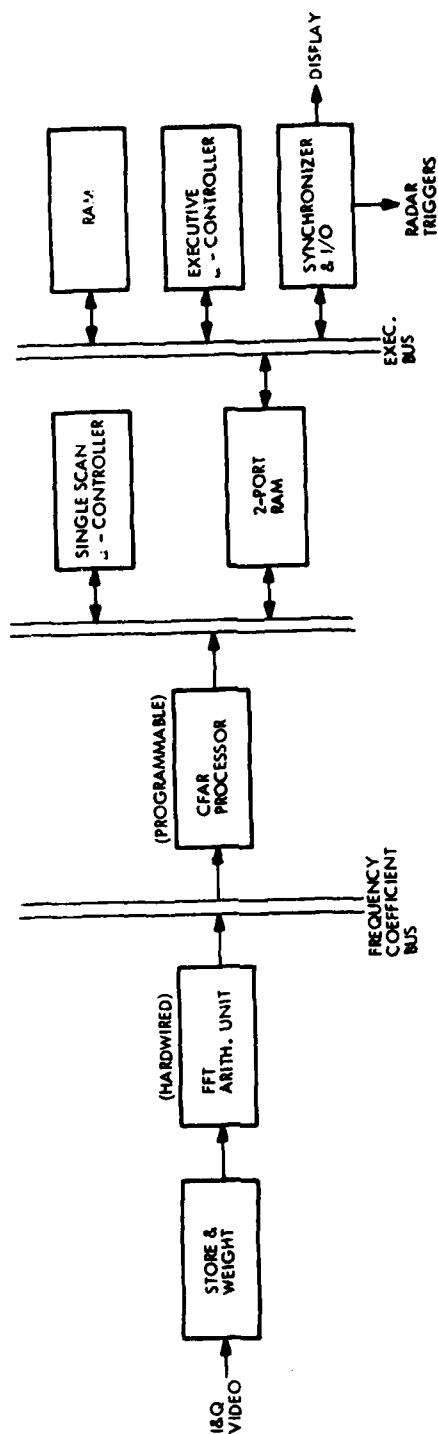


Figure 5-7. Signal Processor Block Diagram  
(Using Universal FFT and CPE CFAR)



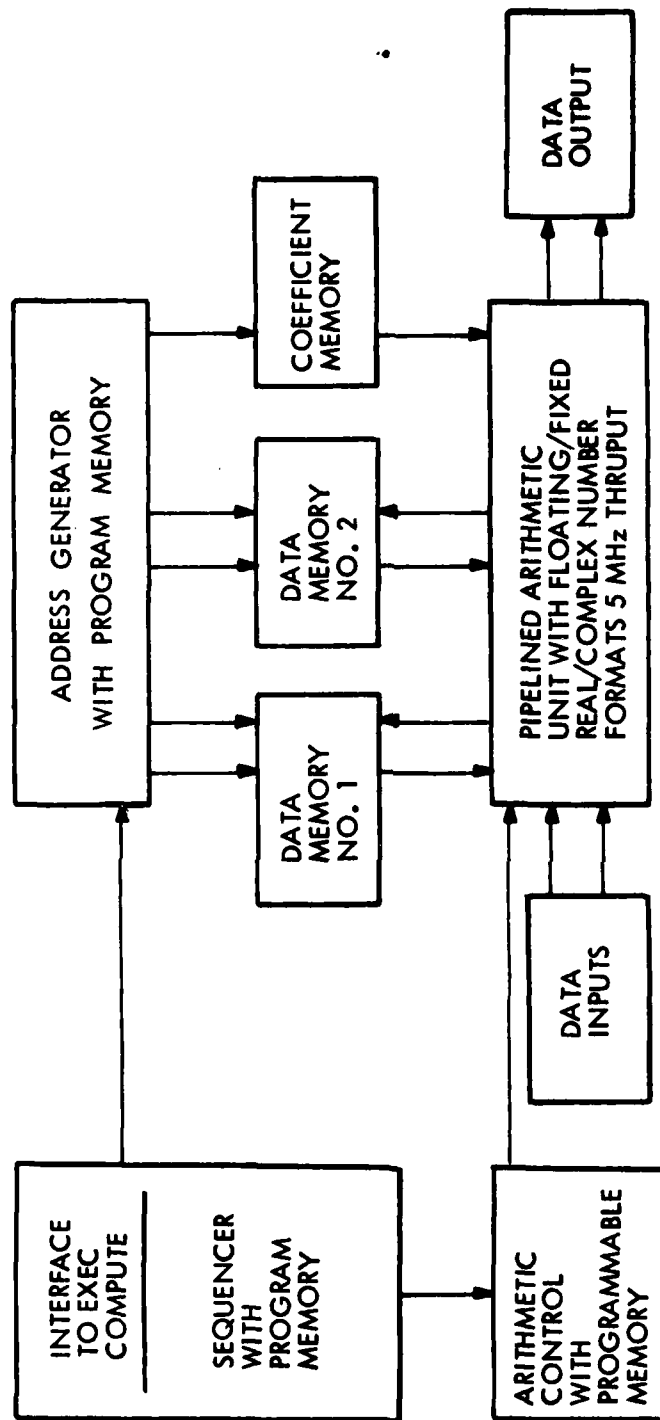


Figure 5-8. Signal Processor: GPSP Architecture

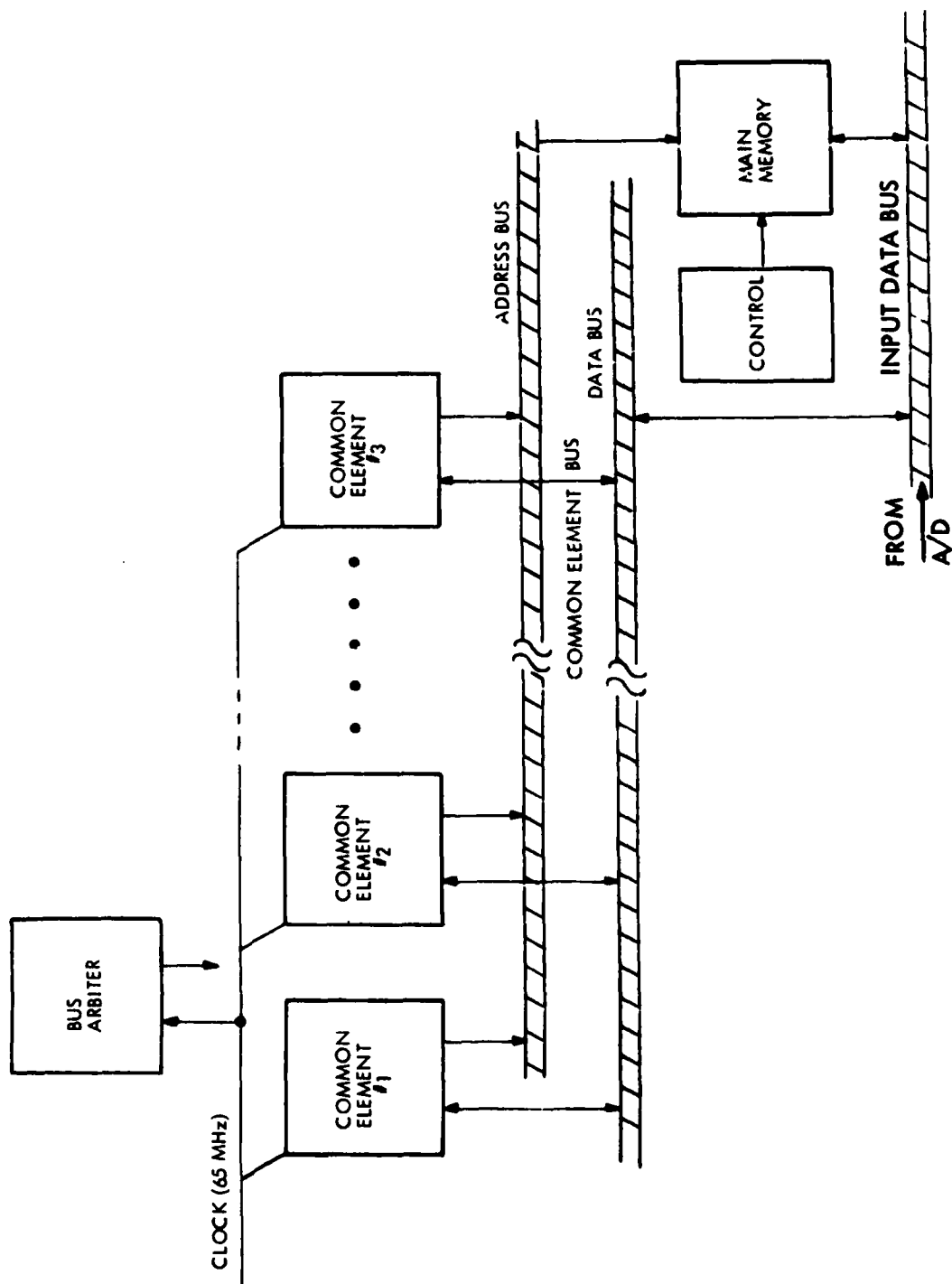


Figure 5-9. Distributed Signal Processor  
(Using Common Elements)

<u>Weighting Process</u>		<u>Operations</u>
LOAD	Q term in Latch	1
MPY	Weighting	12
STO	Q term in place	1
LOAD	I term in Latch	1
MPY	Weighting	12
STO	I term in place	1
.	Insert Butterfly process	
MPY	Weighting (2nd term of Butterfly - I)	12
.		
MPY	Weighting (2nd term of Butterfly - Q)	12
.		
Index Weighting Address		
Total		$\frac{1}{53}$ operations

The total time for a block of N samples at a 5 kHz PRF is  
 $T_p = 2 \times 10^{-4} N$  seconds. Then for R range gates, the operations/sec.  $I_W$  required for weighting is:

$$I_W = 53R/T_p = 265,000 R/N \quad (5-10)$$

#### 5.7.1.2 FFT Processing

The total number of Butterflies B per range cell is:

$$B = \frac{N}{2} \log_2 N = 1.66 N \log_{10} N \quad (5-11)$$

The operations per Butterfly are tabulated below:

FFT Process (Operations)Operations

Using a 16 Sample Batch as an Example

LOAD	$I_8$ in Latch	; Latch	1
MPY	$\cos \theta$	; $I_8 \cos \theta$ in Accum	12
STO	Temp 1	; Temp 1 = $I_8 \cos \theta$	1
MPY	$\sin \theta$	; $I_8 \cos \theta$ in Accum	12
STO	Temp 2	; Temp 2 = $I_8 \sin \theta$	1
LOAD	$Q_8$ in Latch	; Latch	1
MPY	$\cos \theta$	; $Q_8 \cos \theta$ in Accum	12
STO	Temp 3	; Temp 3 = $Q_8 \cos \theta$	1
MPY	$\sin \theta$	; $Q_8 \sin \theta$ in Accum	12
ADD	Temp 1	; Accum = $I_8 \cos \theta + Q_8 \sin \theta$	1
STO	Temp 1	; Temp 1 = $I_8 \cos \theta + Q_8 \sin \theta$	1
LOAD	Temp 3	; Latch	1
SUB	Temp 2	; Accum = $Q_8 \cos \theta - I_8 \sin \theta$	1
STO	Temp 2	; Temp 2 = $Q_8 \cos \theta - I_8 \sin \theta$	1
LOAD	$I_0$	; Latch	1
ADD	Temp 1	; $I'_0 = I_0 + (I_8 \cos \theta + Q_8 \sin \theta)$	1
STO	$I_0$	; Store in original location	1
SUB	Temp 1	; $I_0$ now in Accum	1
SUB	Temp 1	; Accum = $I_0 - (I_8 \cos \theta + Q_8 \sin \theta)$	1
STO	$I_8$	; Store in original location now $I'_8$	1
LOAD	$Q_0$	; $Q_0$ in Latch	1
ADD	Temp 2	; Accum = $Q_0 + (Q_8 \cos \theta - I_8 \sin \theta)$	1
STO	$Q_0$	; Store in original location now $Q'_0$	1
SUB	Temp 2	; $Q_0$ in Latch	1
SUB	Temp 2	; Accum = $Q_0 - (Q_8 \cos \theta - I_8 \sin \theta)$	1
STO	$Q_8$	; Store in original location now $Q'_8$	1
Index	$I_0$		1
Index	$Q_0$		1
Index	$I_8$		1
Index	$Q_8$		1
Total			<u>74</u> operations

As above, we find the total operations as follows:

$$I_F = 74 \frac{\text{ops}}{\text{butterfly}} \times \frac{BR}{T_p N} = 6.14 \times 10^5 R \log_{10} N \quad (5-12)$$

#### 5.7.1.3 Magnitude Processing (per Range Gate)

The magnitude calculation is based on a 3-level approximation algorithm which performs the magnitude calculation based on the values of I & Q from the point of the output of the FFT process. This algorithm produces an output with a peak error no worse at any point than 3.1% of the correct  $\sqrt{I^2 + Q^2}$  value.

Both a table of operations and a flow chart of the magnitude processing algorithm are shown below.

	<u>Magnitude Process</u>			<u>Operations</u>
	LOAD	I	; I in Accum	1
	SUB	Q	; I-Q in Accum	1
	BMI	SWAP	; I < Q?	1
	LOAD	I		1
	STO	Temp 1	; store larger	1
	LOAD	Q		1
LARG:	STO	Temp 2	; store smaller	1
	SHR		; shift right	1
	SHR		; divide by 4	1
	ADD	Temp 1		1
	STO	Temp 3		1
	LOAD	Temp 1		1
	SHR			1
	STO	Temp 1	; C/2	1
	SHR			1
	ADD	Temp 1		1
	STO	Temp 4	; 3/4 C	1
	SUB	Temp 2	; 3/4 C-D	1
	BMI	EIT	; D > 3/4 C	1
	LOAD	Temp 1	; C/2	1
	SUB	Temp 2	; C/2-D	1
	BMI	THTO	; D > C/2	1
	LOAD	Temp 3	; Load magnitude	1
FINAL:	STO	in I	; store in I	1
	EXIT			1
SWAP:	LOAD	Q	; Q in Accum	1
	STO	Temp 1	; store larger	1
	LOAD	I	; load smaller	1
	BR	LARG	; return	1
EIT:	LOAD	Temp 2	; load smaller (D)	1
	SHR			1
	SHR			1
	SHR		; divide by 8	1
	ADD	Temp 3	; mag + D/8	1
	BR	FINAL		1
THTO:	LOAD	Temp 2	; load smaller, D	1
	SHR			1
	SHR			1
	SHR			1
	SHR			1
	SHR		; divide by 32	1
	ADD	Temp 3	; mag + D/32	1
	BR	FINAL		1
EXIT:	Index I			1
	Index Q			1
Worst Case Total				34 operations

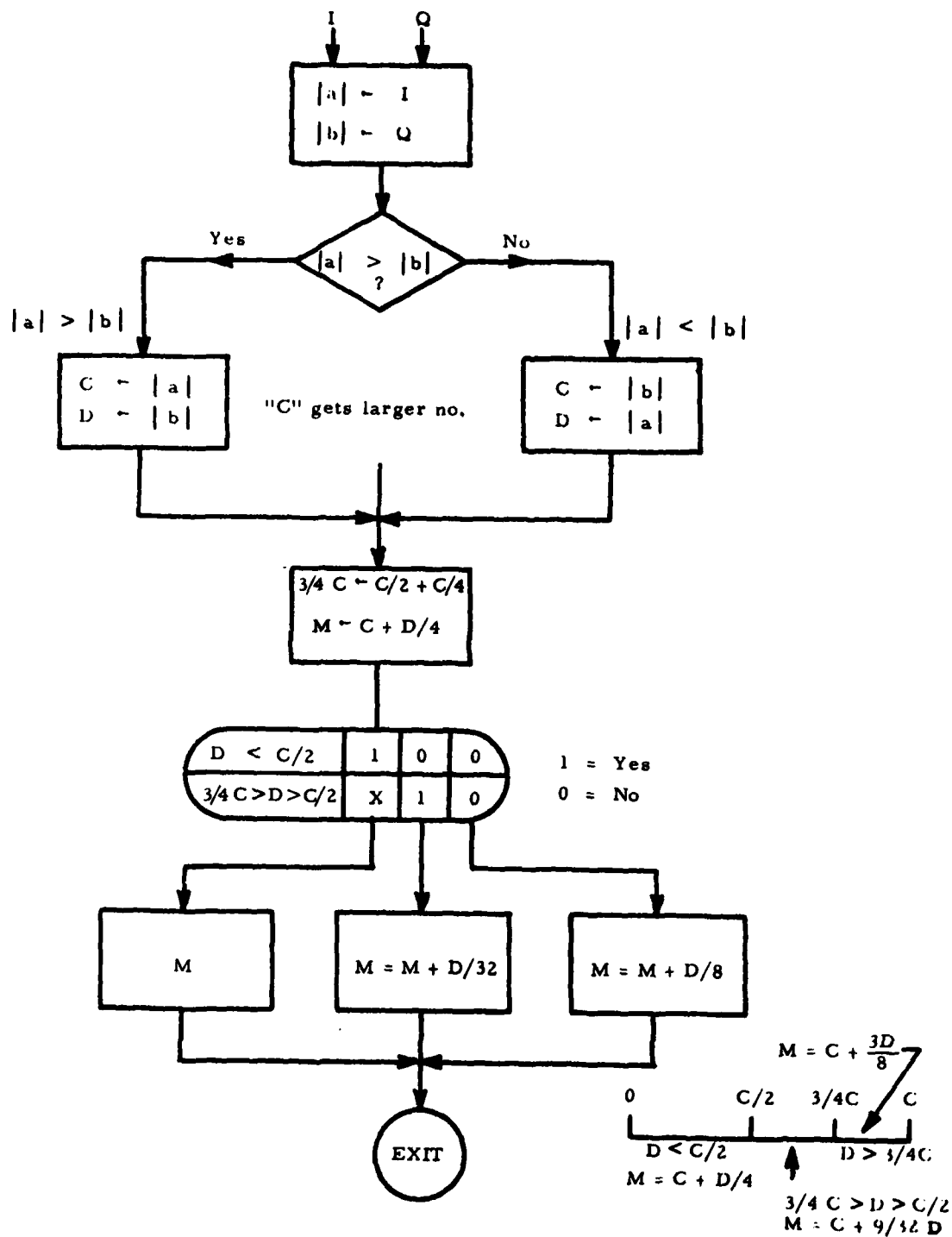


Figure 5-10. Magnitude Determination Flow Chart

The magnitude operations per second are independent of block size, and is given by:

$$I_M = 34 NR / (T_p N) = 170,000 R \quad (5-13)$$

#### 5.7.1.4 Range Averaging CFAR

The Range Average processing is based upon summing 16 range cells before the range cell of interest and 16 summing range cells after the range cell of interest. A range cell space is inserted on both sides of the range cell of interest and the two sums. A tabulation of operations follows:

<u>Range Average Process</u>			<u>Operations</u>
BR	No op	; if count less than 16	1
LOAD	$S_{n-1}$		1
SUB	$R_n$		1
ADD	$R_{n+1}$		1
STO	$S_n$	; late sum	1
CMP	$S_{n-19}$	; compare both sums	1
BR	LARG		1
LARG: LOAD	SUM	; load larger sum	1
SHR			1
SHR			1
SHR			1
SHR		; divide by 16	1
STO	THRESHOLD	; store in threshold	1
INDEX	$S_n$		1
INDEX	$S_{n-19}$		1
INDEX	$R_{n+16}$		1
INDEX	COUNT	; total number of range gates	1
BR	COUNT GREATER		1
Total			<u>1</u> 20 operations

Operation rate here is given by:

$$I_{RA} = 1000,000 R$$



### 5.7.1.5 Frequency Averaging CFAR

This is basically a broad-band interference reduction technique. It is accomplished by summing the magnitude from all the filters of the output of the FFT Process and dividing by the total number summed. Tabulating operations:

<u>Frequency Average</u>			<u>Operations</u>
(for a 64 point transform)			
LOAD	$f_0$		1
ADD	$f_1$		1
•			•
•			•
•			•
ADD	$f_{63}$	; Accumulate 64 terms	1
SHR	SUM		1
SHR	SUM		1
SHR	SUM		1
SHR	SUM		1
SHR	SUM	; divide by 64	1
SHR	SUM		1
CMP	THR	; compare to threshold	1
STO	LARGER		1
Total			72 operations

Operation rate  $I_{FA} = 6875R$

### 5.7.1.6 Thresholding

This operation is based upon the following computation: four thresholds are compared (clutter, range average, frequency average and minimum threshold), the largest is selected and multiplied by a CFAR constant, then compared to the magnitude of the range gate of interest.

<u>Threshold Process</u>			<u>Operations</u>
LOAD	Clutter		1
CMP	Range		1
BR	Larger		1
CMP	Freq.		1
BR	Larger		1
CMP	Min.		1
BR	Larger		1
MPY	CFAR Constant		12
CMP	Target		1
FORMATTING			20
Total			40 operations

$I_{TH} = 200,000 R/N$

The total of all of the above operation rates is given by:

$$I_{TOT} = R \left[ 276,875 + \frac{465,000}{N} + 6.14 \times 10^5 \log N \right] \quad (5-14)$$

Given a card implemented with high-speed micrologic which is capable of processing 16.67 MIP ( $16.67 \times 10^6 = 1$ ), the number of cards required for all operations is:

$$C_{CE} = \text{INT} \left( \frac{I_T}{16.67 \times 10^6} \right) + 1 \quad (5-15)$$

Since one card is needed for control, etc.

### 5.7.2 The GPSP Approach

Since this is an existing design whose properties are known, we do not need to approach it analytically as above. We simply note that 50 cards as a unit can process  $10^7$  butterflies/sec., as well as handling the other needed operations. But the required number of butterflies/sec. is:

$$B_{REQ} = \frac{B}{T_p} = 8.3 \times 10^3 R \log N \quad (5-16)$$

The GPSP can only be used in modules of 50 cards. So the card count here is given by:

$$\begin{aligned} C_{GPSP} &= 50 \left[ \text{INT} \left( \frac{B_{REQ}}{10^7} \right) + 1 \right] \\ &= 50 \left[ \text{INT} ( 8.3 \times 10^{-4} R \log N ) + 1 \right] \end{aligned} \quad (5-17)$$

### 5.7.3 The UFFT/CPE Approach

For the UFFT, 12 cards process  $1.67 \times 10^6$  butterflies/sec., so as above:

$$C_{UFFT} = 12 \left[ \text{INT} \left( 4.97 \times 10^{-3} R \log N \right) + 1 \right] + 2 \quad (5-18)$$

(The 2 represents array multiplier cards needed for weighting.)

The CPE total instructions per second are:

$$I_{TCPE} = I_{RA} + I_{FA} + I_{TH} \quad (\text{calculated as in 5.7.1})$$

$$= R \left[ 106,875 + \frac{200,000}{N} \right] \quad (5-19)$$

Since the CPE processes  $3.3 \times 10^6$  operations/sec., and requires about 1 control card for each six processing cards:

$$C_{CPE} = \text{INT} \left( \frac{I_{TCPE}}{3.3 \times 10^6} \right) + \text{INT} \left( \frac{I_{TCPE}}{2 \times 10^7} \right) + 2 \quad (5-20)$$

The total card count for the UFFT/CPE approach is the sum of  $C_{UFFT}$  and  $C_{CPE}$ .

#### 5.7.4 The Input Buffer

The memory requirement is given by:

$$m = R \times N \times 2 \text{ (for I \& Q)} \times 2 \text{ (for double buffering)}$$

$$= 4RN \text{ words}$$

Assume that the most that we can put on one card is 2048 words (16-bit words, 1024 x 1 memory IC's, 32 memory IC's per card plus some control circuitry). Then the number of buffer cards needed is:

$$C_{IB} = \text{INT} \left( \frac{RN}{512} \right) + 1 \quad (5-21)$$

#### 5.7.5 The Clutter Map

Assume 50 angle cells ( $100^\circ$  scan in  $2^\circ$  steps) and  $R$  range gates, with 16 bits of averaged clutter amplitude information stored per cell. (This is probably too much, and will lead to a conservative estimate.) Then with 2048 words stored per card, the number of cards needed for clutter map memory alone is:

$$C_{CM} = \text{INT} \left( \frac{50R}{2048} \right) + 1 \quad (5-22)$$

The clutter map calculation, for a decaying exponential clutter memory, will follow this algorithm.

$$S_{n+1} = K S_n + |R_i| \quad \text{where } K = 1-2^{-j} \quad (5-23)$$

Where  $R_i$  is the clutter amplitude of a newly arriving sample,  $S_n$  is the stored clutter sum, and  $K$  is an exponential weighting constant. The operations are tabulated below:

<u>Clutter Map Process</u>				<u>Operations</u>
LOAD	$S_n$	; old value		1
SHR		; divide by 64		1
SHR				1
SHR				1
SHR				1
SHR				1
SHR				1
SUB	$S_n$	; old value		1
ADD	$R_i$	; add magnitude of range cell		1
STO	$S_n$	; new value of $S_n$		1
INDEX		; update for new sum		1
Total				11 operations

The map processing load is a function of the scan time  $T_o$  (nominally 0.5 second).

$$I_{\text{CMPROC}} = 50 \text{ angle cells} \times R_{\text{range cells}} \times 11_{\text{ops}} / 0.5 \text{ sec.} \quad (5-24)$$

$$= 1100R \text{ ops/sec.}$$

Assuming that the CPE approach (3.3 MIPS) is used for implementing this algorithm, we have approximately:

$$C_{\text{CMPROC}} = \text{INT} \left( \frac{1100R}{3.3 \times 10^6} \right) + 1 \quad (5-25)$$

#### 5.7.6 Other--The Synchronizer

Six (6) cards are assumed for the synchronizer, common to all approaches.

### 5.7.7 Hardware Comparisons

A comparison of hardware cost alone (and we can consider this as being proportional to total card count) is of interest in that it is a measure of the relative cost of signal processors beyond the first model (within which are concentrated development and software costs). Figure 5-11 gives an example of the relative "cost" measured in cards, of a processor with  $N = 64$ , for varying range cell number, and for the three processor approaches considered here. The Common Element approach appears consistently to cost less, although not always by a significant amount. These curves have been plotted from the equations presented in the preceding paragraphs. Another useful way of looking at these results appears in Figure 5-12; here we consider the situation of a constant cost system ( $C = 100$ ), and examine what performance is available (measured by range gates  $R$  and points  $N$ ) using the three approaches. Again the CE approach has a slight edge except where the number of range gates becomes of the order of 1000 or more, where the GPSP appears to gain an advantage.

### 5.8 Development Requirements

#### 5.8.1 Software

The various operation sequences tabulated in the preceding paragraphs do more than place a demand on the hardware in terms of operations per second. They are programs, and must be written and debugged for all of the approaches considered. Summarizing the lines of code that need to be programmed:

Input Weighting	9	
FFT Operation	30	
Magnitude	45	
Frequency Average	72	
Clutter Threshold	11	
Range Average	20	
Thresholding	40	
	<u>227</u>	
Multiply Subroutine	14	
	<u>241</u>	
50% Margin	120	
Total	<u>361</u>	lines
	<u>x 4</u>	each line equivalent to four
	1444	16-bit lines

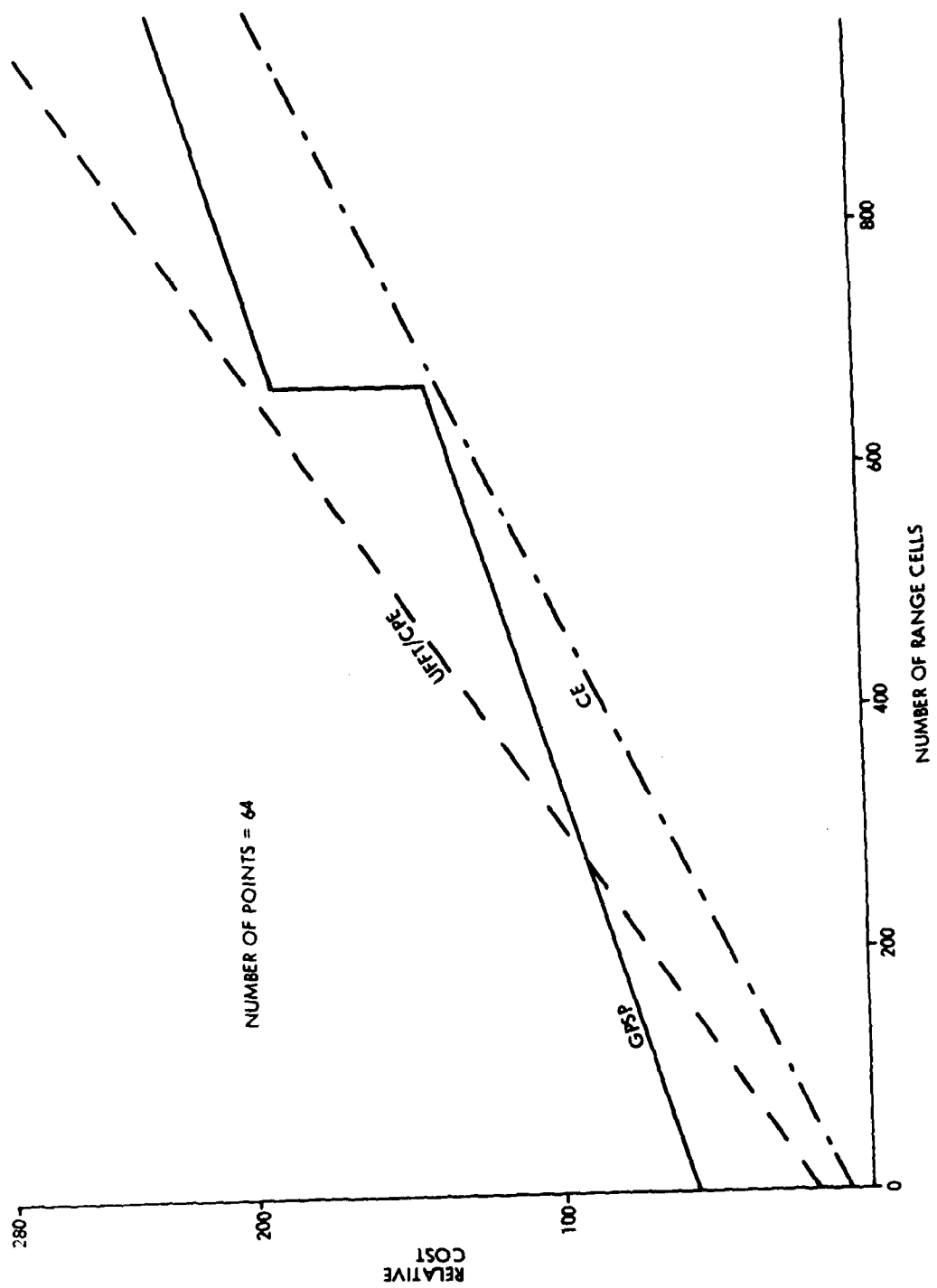


Figure 5-11. Cost Comparison: Hardware Only

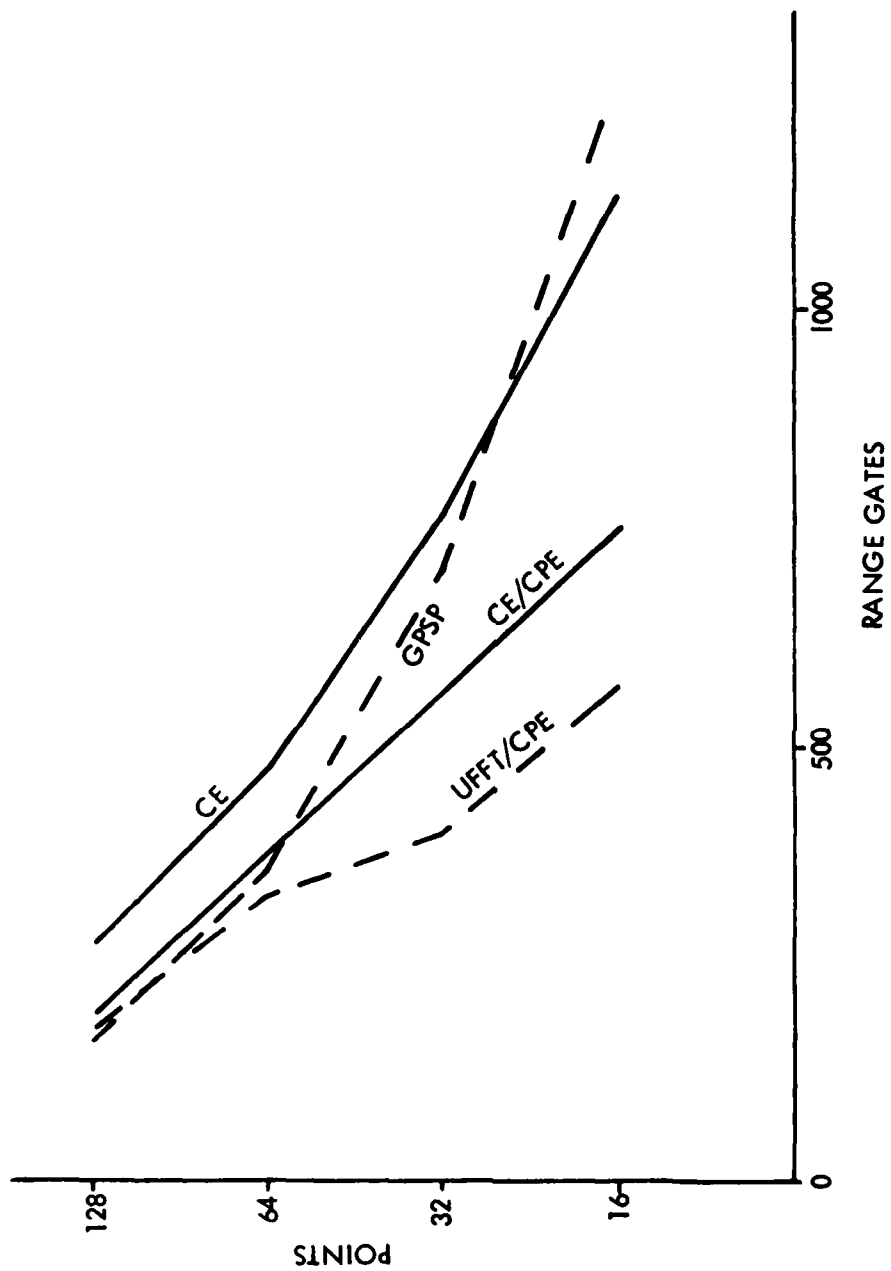


Figure 5-12. Performance Available for Constant Hardware Cost ( $C = 100$ )

Assuming that each fully documented and debugged line represents the expenditure of 1 engineering man-hour, the total software cost for all approaches is about 200 man-days.

#### 5.8.2 Other Costs

(a) **Hardware Development:** Even when card designs exist, they must be packaged, interconnected, furnished with power supplies and controls, etc. For the Common Element (CE) approach, the card designs are new and must also be accomplished.

(b) **System Design and Test:** This includes working out many system design details, writing appropriate unit and interface specifications, and testing the finished unit (which usually also implies the design of special testing hardware). This also includes documentation of the finished unit.

The hardware cost data of the preceding section was developed in terms of equivalent "cards". Most of the development fixed costs are basically engineering manpower costs. So as to be able to use the original curves without modification, conversion factors are used to relate "cards" to man-days. Using the neutral "card" unit seems permissible because relative costs are of prime interest here. Also, using dollars can be misleading because of inflationary factors. The basic equivalence is 1 tested card = 8 engineering man-days.

Summarizing the fixed development costs for the three approaches (and it must be realized that because no design detail is available, these estimates are very rough):

	<u>Software</u>	<u>Hardware Design</u>	<u>Systems</u>	<u>Total</u>	<u>Equivalent Cards</u>
GPSP	200 m-days	120 m-days	300 m-days	620 m-days	80
UFFT/CPE	200	120	300	620	80
CE	200	500	400	1100	135

We can combine the variable hardware costs with the fixed development costs to form the curves of Figure 5-13 and 5-14. Note that the GPSP looks more attractive here, because of the higher development cost associated with the CE approach.



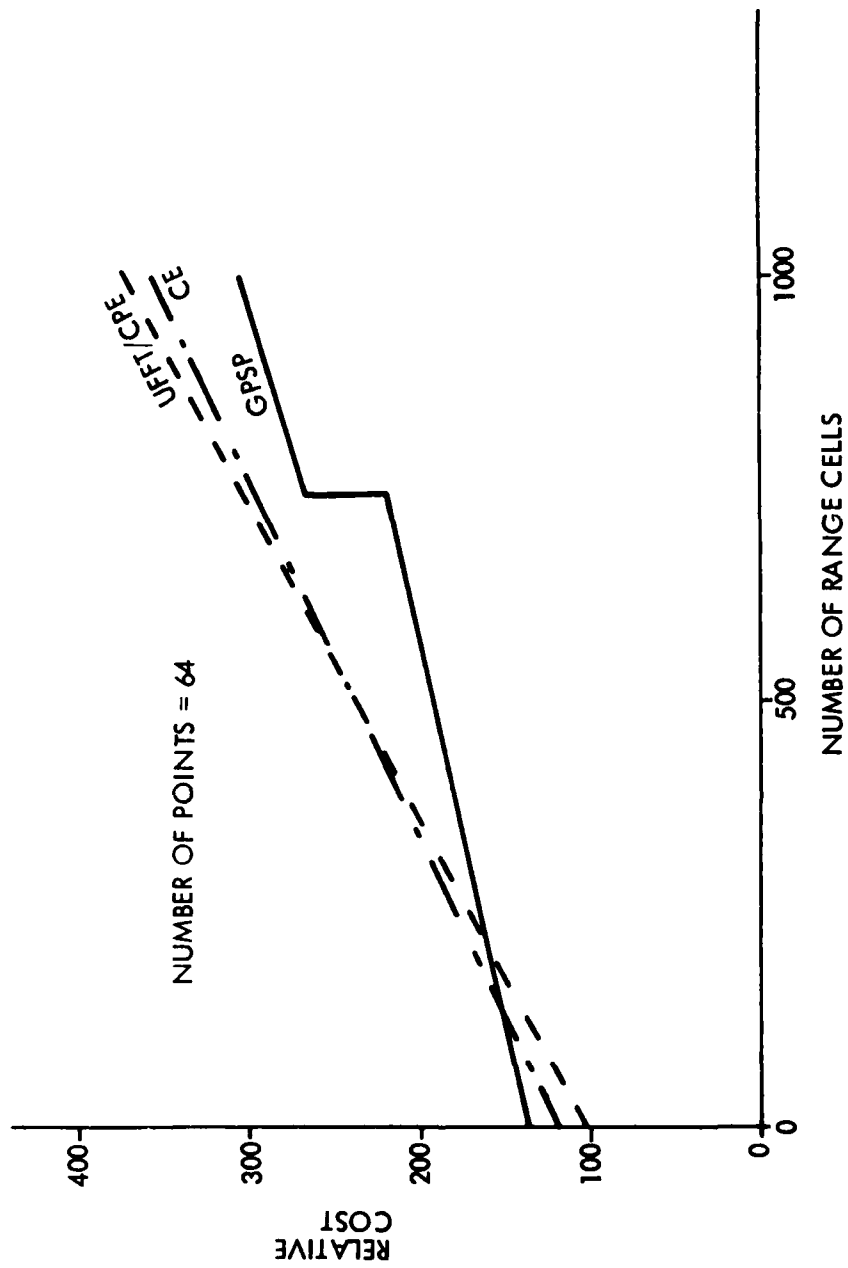


Figure 5-13. Cost Comparison: Hardware Plus Development Costs

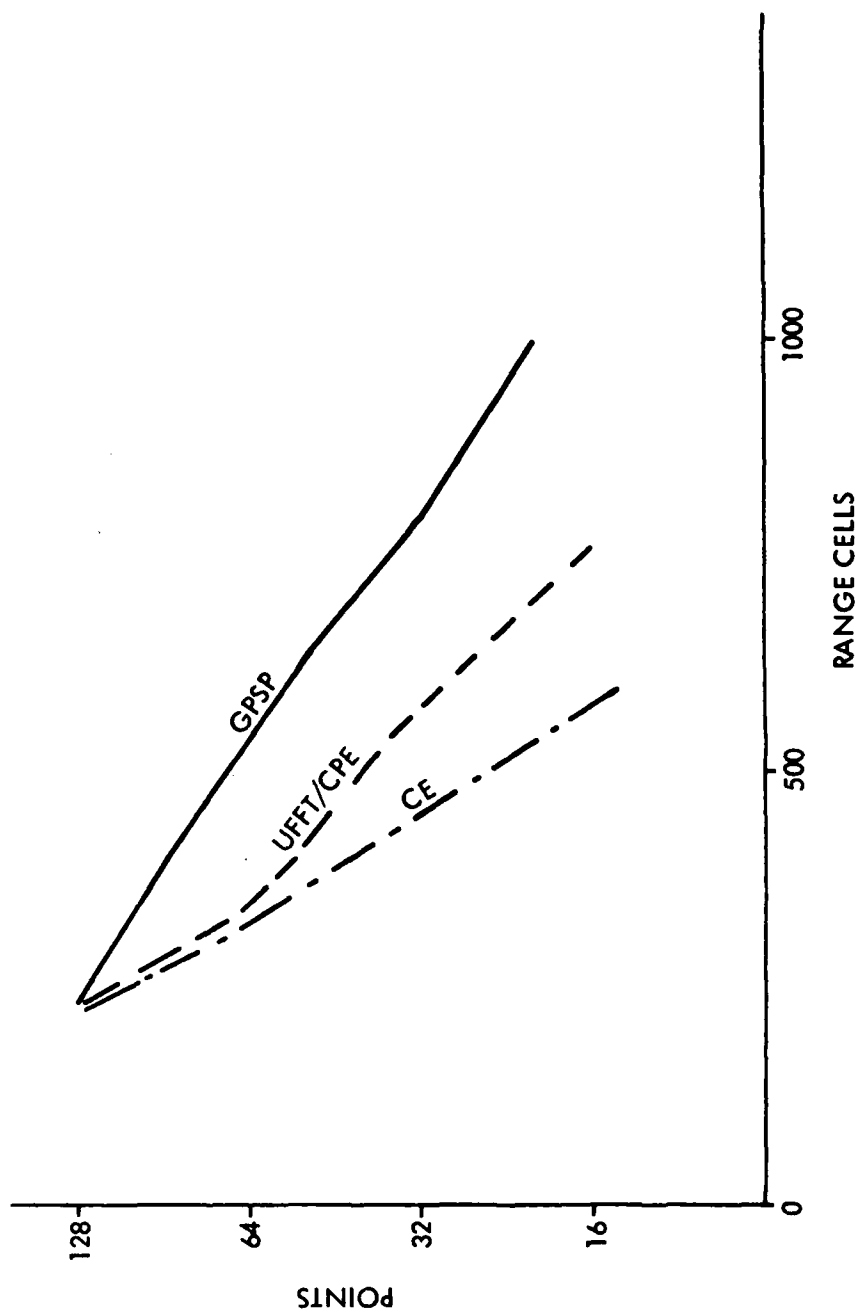


Figure 5-14. Performance Available for Constant Total Cost ( $C = 200$ )

## SECTION 6. CONCLUSIONS AND RECOMMENDATIONS

6.1 The elements of a CFAR signal processor have been defined and described in some detail. It is clear that the coherent processing offered by the FFT will improve radar sensitivity, and that the CFAR circuitry suggested here--range-cell averaging, clutter mapping--should suppress common types of false target sources.

6.2 The limits on processing will be set ultimately by cost. Some insights have been gained into the relative costs of various processing approaches, as well as the relationship between cost and performance parameters such as number of pulses coherently integrated and number of range cells handled.

6.3 Further refinement of the analysis would be desirable. This is particularly true in the following areas:

(a) MTI: We have investigated only two MTI approaches: three-pulse MTI and narrow-notch N-pulse MTI. Neither is felt to be optimum for the EAR environment.

(b) Clutter-map parameters: More data would be desirable on the effect of clutter-map parameters (i. e., the number and "age" of clutter samples averaged and stored) on system performance.

(c) Quantization: Although the present results are good enough to be used with the present 9-bit EAR A/D converter, no analysis exists to tell us whether anything would be gained if (at some future time) it were decided to increase the number of bits in the processor.

(d) Coherent versus Non-Coherent Integration: For a given beam position dwell time, there is an interesting trade-off to be examined between coherent integration (FFT) and incoherent integration, with the possibility of rf frequency changing between coherent pulse groups taken into account

as a means of reducing target fluctuation losses. This tradeoff affects sensitivity; the effect of such integration changes on CFAR performance, which is more complex, also needs to be taken into account.

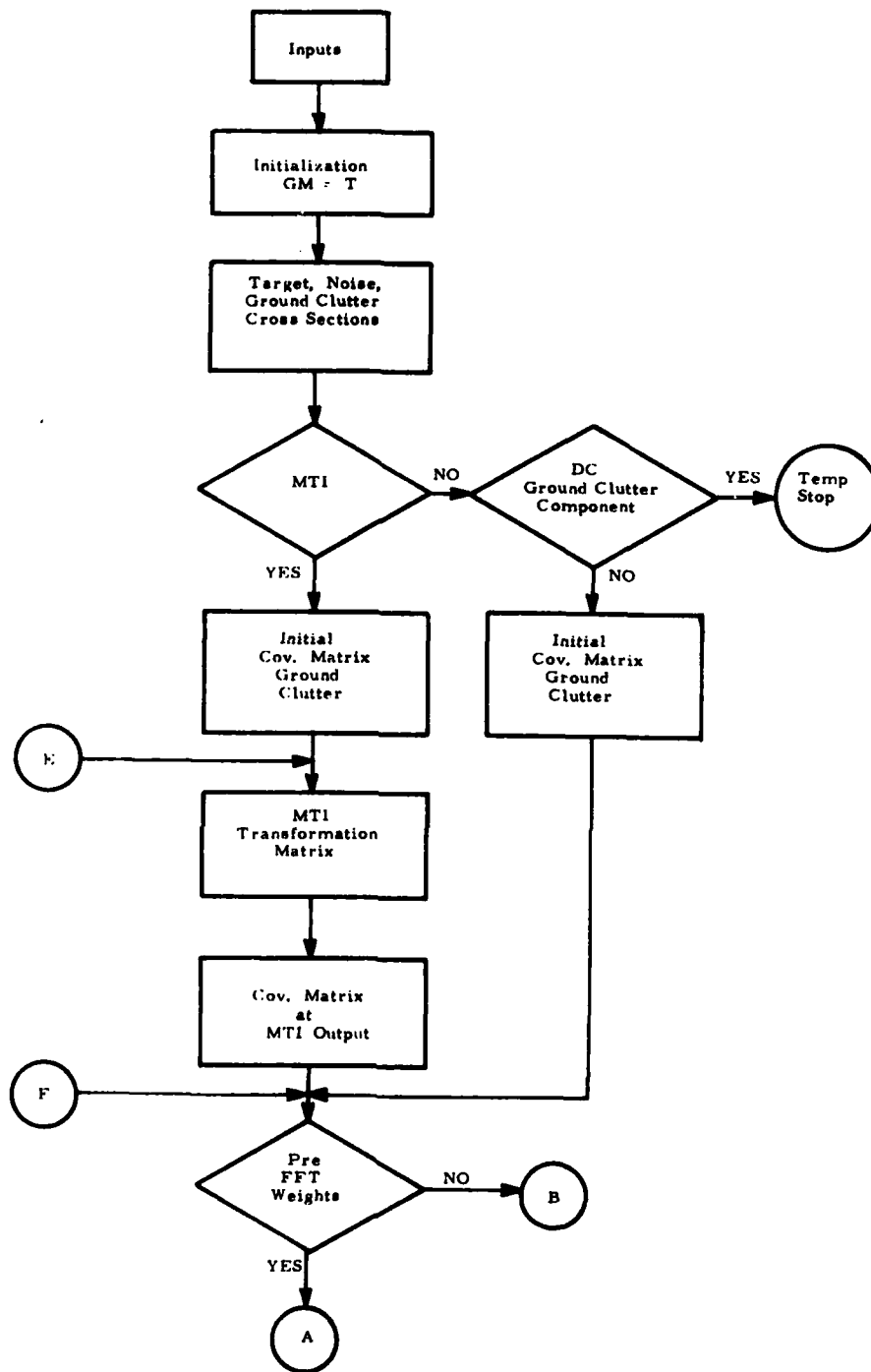
6.4 In the hardware area, it is clear that, although a substantial beginning has been made, the level of detail is insufficient for a design to be initiated. But we have identified three usable processor techniques (CE, GPSP, and UFFT). The Common Element (CE) approach is the newest, and as a result entails more development cost. But it has the promise of least ultimate hardware cost, as well as being an inherently modular design, with great flexibility in terms of both re-programmability and growth to more range cells/samples. A development program can be visualized which starts out modestly with the development of only one Common Element module, upon which various processing algorithms can be exercised. Ultimately, the modules can be replicated in sufficient quantity to form a total processor of whatever capability is desired.

## REFERENCES

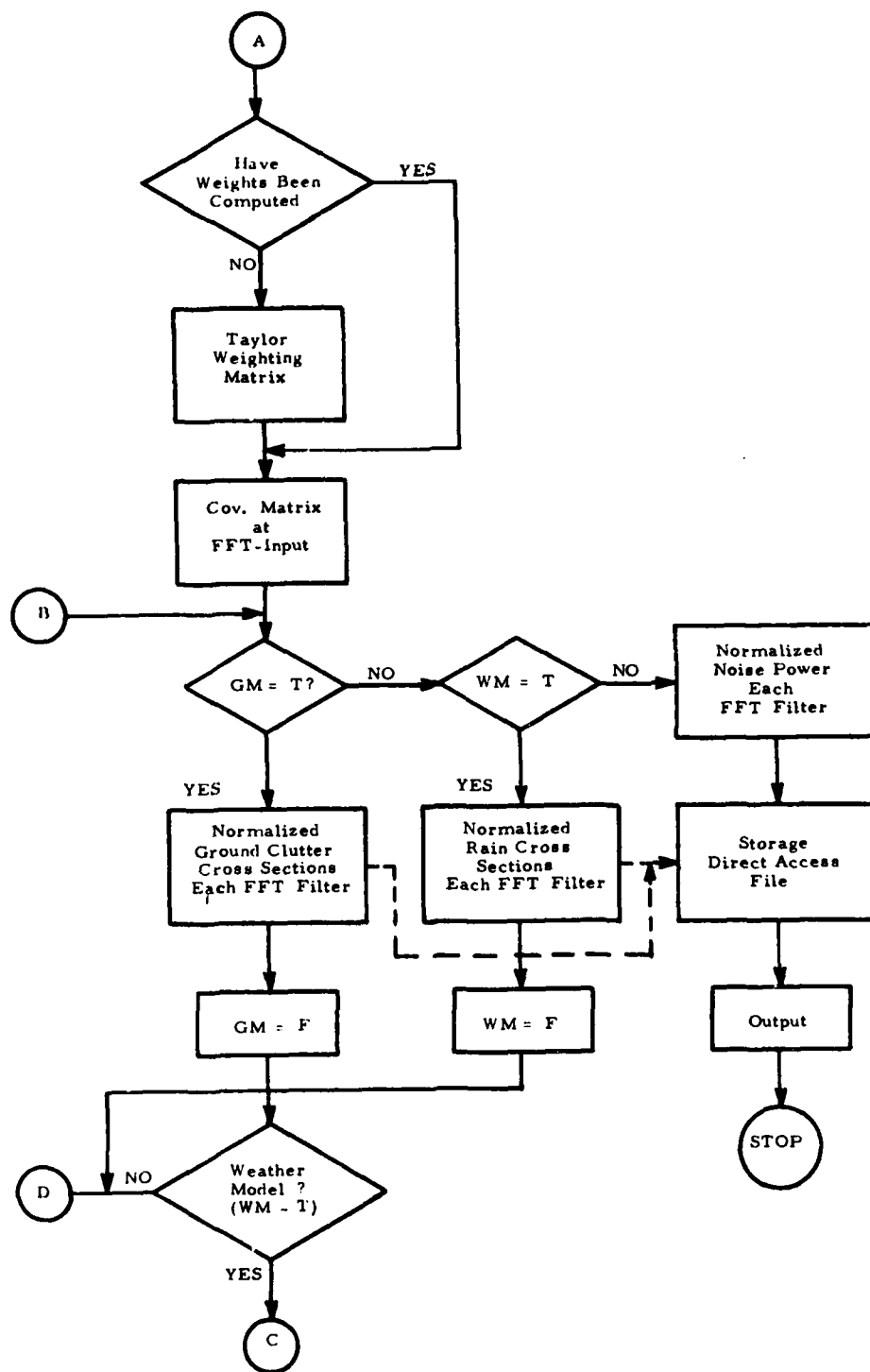
1. "Experimental Array Radar (EAR) Second-Phase Development Plan," J. L. Hatcher, Advanced Sensors Directorate, MICOM. Tech. Report RE-73-12, 1 March 1973.
2. "MICOM Experimental Array Radar Digital Signal Processor," J. F. Lancaster, Advanced Sensors Directorate, MICOM. Tech. Report RE-73-14, 22 February 1973.
3. "Radar System Analysis," D. K. Barton, (Prentice-Hall).
4. "Radar Design Principles," F. E. Nathanson (McGraw-Hill).
5. "A Model Environment for Search Radar Evaluation," H. R. Ward, EASCON '71 Convention Record.
6. "Detection Performance of the Cell-Averaging LOG/CFAR Receiver," V. G. Hansen and H. R. Ward, IEEE Trans. Vol. AES-8, PP 648-652, September 1972.
7. "On the Detection Performance of a Cell-Averaging CFAR in Non-Stationary Weibull Clutter," L. Novak, (1974 IEEE Symposium on Information Theory).
8. "The Weibull Distribution Applied to the Ground Clutter Backscatter Coefficient," R. R. Boothe, U.S. Army Missile Command, Redstone Arsenal, RE-TR-69-15, 12 June 1969.
9. "Analysis of the Correlation of a Single Set of Land Clutter Data," F. E. Nathanson and L. W. Brooks, Technology Service Corporation, Memo TSC-W2-19, May 1973.
10. "Spatial Correlation of Land Clutter," F. E. Nathanson and L. W. Brooks, Technology Service Corporation, Memo TSC-LB-jab, July 1973.
11. "Radar Target Detection," D. P. Meyer and H. A. Mayer, (Academic Press).

APPENDIX A

PROGRAMS AND FLOW CHARTS

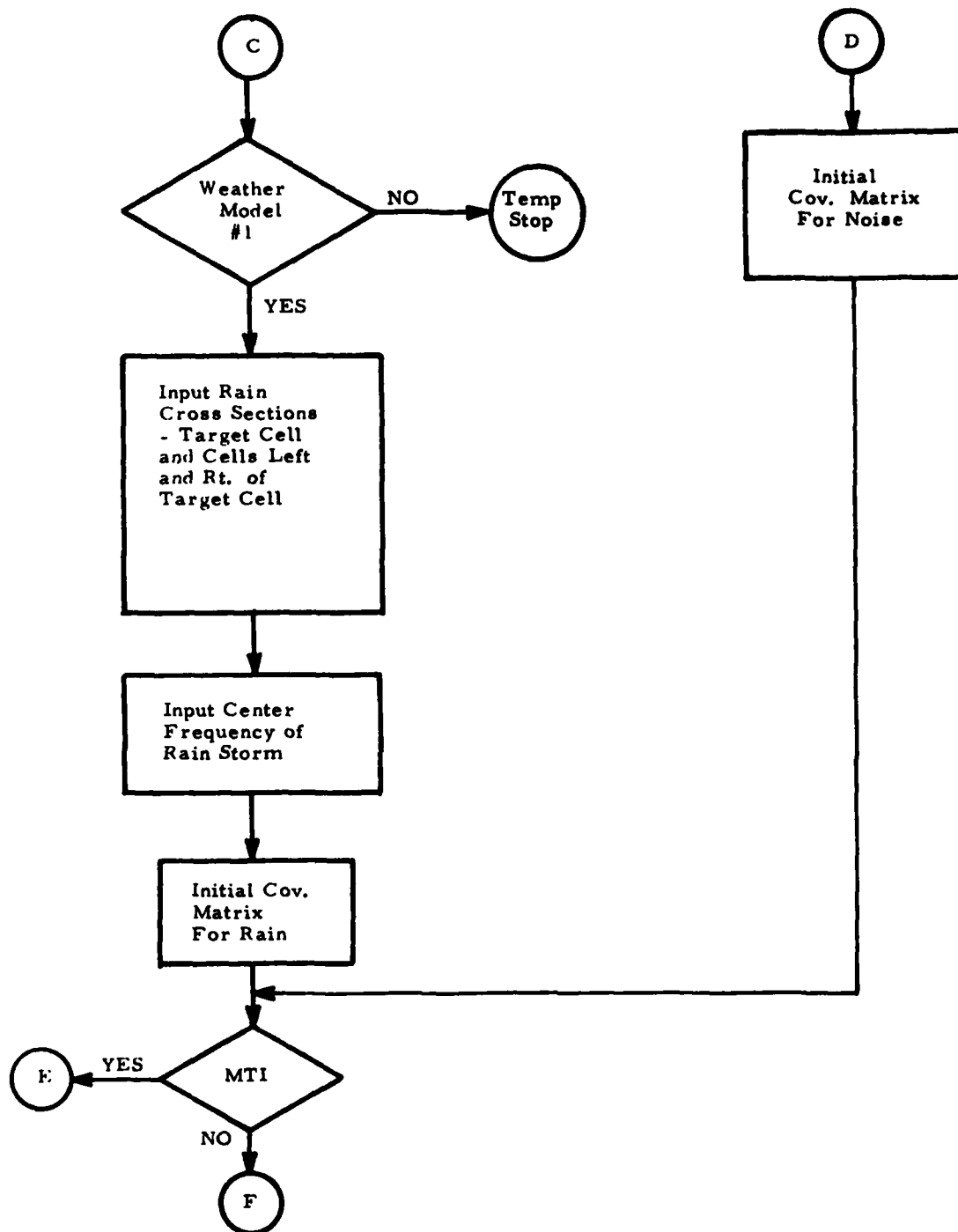


Program CFAR Flow Chart - Sheet 1

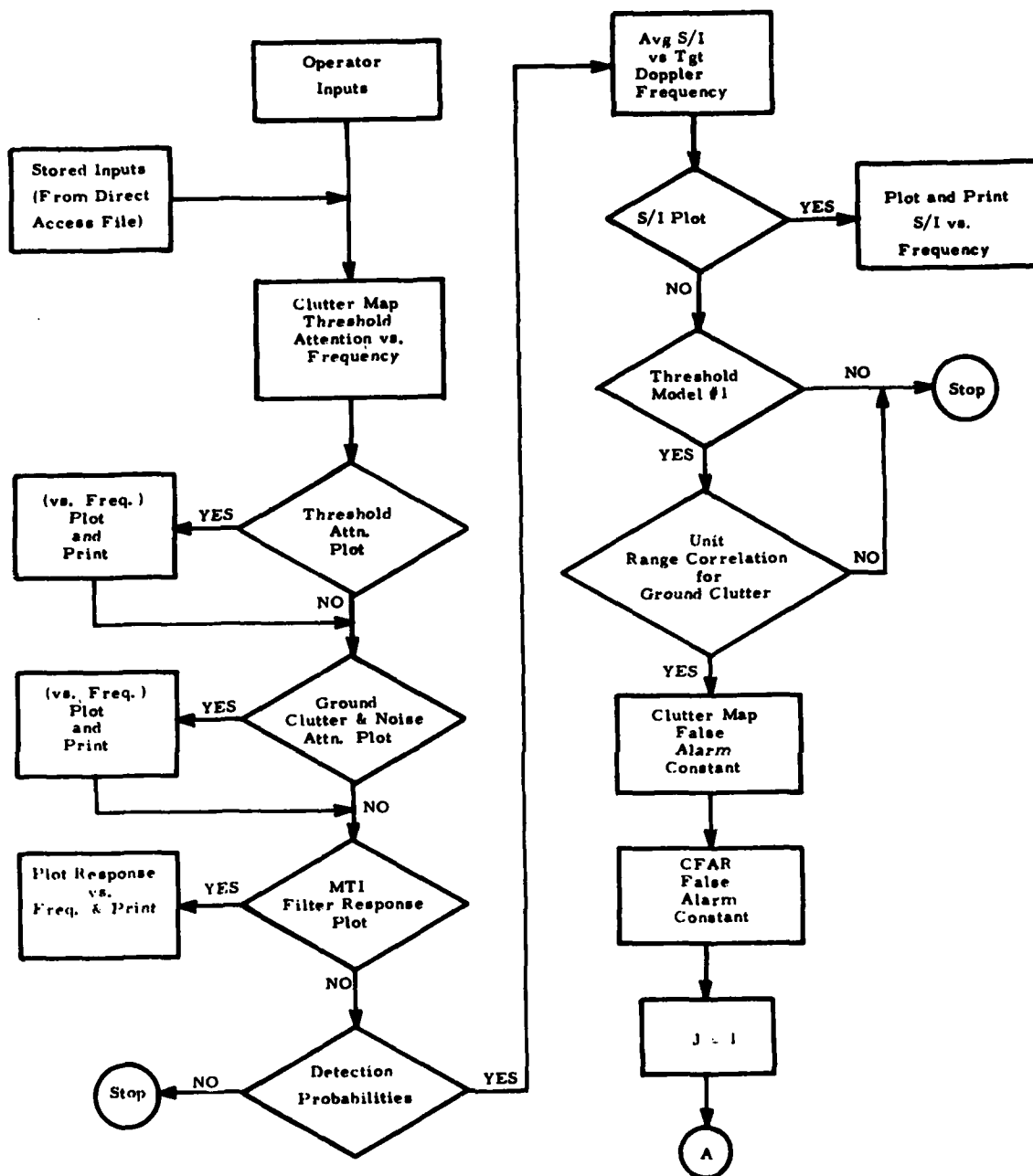


Program CFAR Flow Chart - Sheet 2

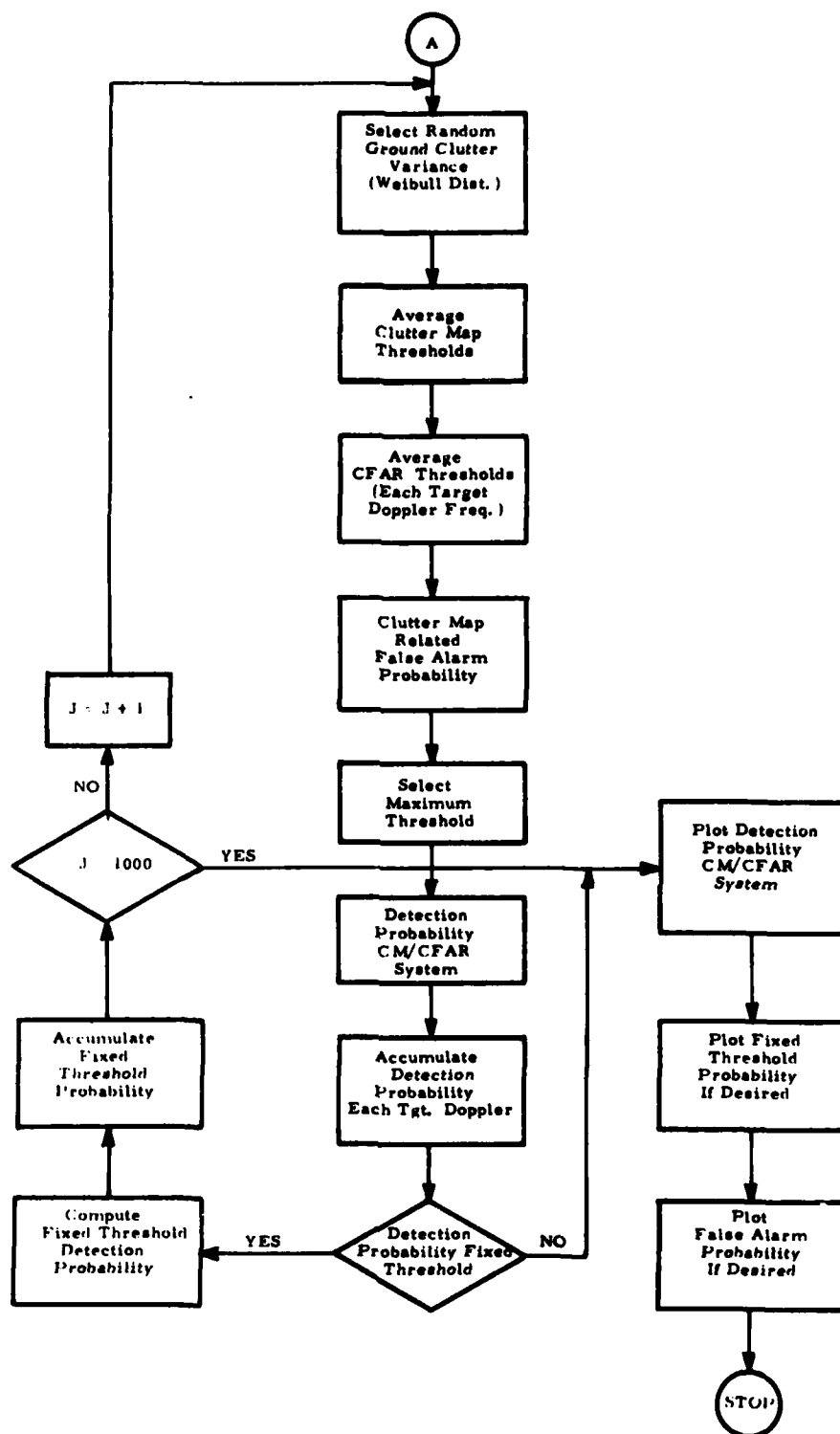




Program CFAR Flow Chart - Sheet 3



Program CFAR Flow Chart - Sheet 4



Program CFAR Flow Chart - Sheet 5

PROGRAM CFAR(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE2)

```

C
C PROGRAM CFAR WRITTEN FOR P.CORNWELL BY C.S.COREY, APRIL 1976
C
C
C INPUTS
C
C NPTS-NUMBER OF POINTS IN FFT
C RHOG-GROUND CLUTTER CORRELATION COEFFICIENT
C NRCELLS-NUMBER OF RANGE CELLS
C WMDL-WEATHER MODEL #1 OR WEATHER MODEL #2
C MTI-TURNS MTI FILTER ON OR OFF
C MTINDL-MTI MODEL #1 OR MTI MODEL#2
C THMDL-THRESHOLD MODEL #1 OR THRESHOLD MODEL #2
C NCLMAP-NUMBER OF POINTS IN CLUTTER MAP
C STNDB-SIGNAL TO NOISE RATIO IN DB
C STBCDB-SIGNAL TO GROUND CLUTTER RATIO IN DB
C A-WEIBULL SLOPE PARAMETER
C T
C IWGT-TURNS WEIGHTS ON OR OFF
C SIGSQ-SPECTRAL WIDTH OF GROUND CLUTTER
C SIGSW-SPECTRAL WIDTH OF WEATHER CLUTTER
C GH-TURNS GROUND CLUTTER MODEL ON OR OFF
C NM-TURNS NOISE MODEL ON OR OFF
C WM-TURNS WEATHER MODEL ON OR OFF
C MUW1-MEAN WEATHER SIGNAL TO LEFT OF CELL OF INTEREST
C MUW2-MEAN WEATHER SIGNAL TO RIGHT OF CELL OF INTEREST
C FW-CENTER FREQUENCY FOR RAIN STORM
C MUW0-MEAN WEATHER SIGNAL CELL OF INTEREST
C KCLUT-CONSTANT CLUTTER FLAG
C STKCLDB-SIGNAL TO CONSTANT CLUTTER IN DB
C NBAR-PARAMETER FOR TAYLOR WEIGHTS CALCULATION
C ATAY-PARAMETER FOR TAYLOR WEIGHTS CALCULATION
C ITATTN-PLOT OPTION FOR CLUTTER MAP THRESHOLD ATTENUATION VS.
C   FREQUENCY PLOT
C IATTN-PLOT OPTION FOR GROUND CLUTTER ATTENUATION VS. FREQUENCY PLOT
C MORCFAR-
C
C LOGICAL MTI,GH,WM,KCLUT,IWGT,NM,ITATTN,IATTN,MORCFAR
C
C REAL MUT,MUN,MUCL,KCLTS,KCL,MUCLF,MUW0,MUW1,MUW2
C
C INTEGER CHECK,WMDL,THMDL,WFLAG
C
C DIMENSION WGT(64),R(66,66),B(66,66),PZ(66,66),
1 BT(66,66),PZTEMP(66,66),PZTIL(64,64),C(64,64),CT(64,64),
2 PW(64,64),CP(64,64),D(64,64),SIGCSQ(64),SIGWSQ(64),SIGNSQ(64)

```

```

C      EQUIVALENCE (CT,BT,CP,ATTN,TATTN,TI,STIDB),(D,R,ATTNDB,TATTNDB)
C      EQUIVALENCE (B,PW)

C      DATA PI/3.14159265/,T/2.0E-04/
C      DATA RHOG,NRCCELLS,WMDL,MTI/1.0,4,1,.T./,MTIMDL,THMDL,
1      NCLMAP,STNDB/1,1,4,10.0/,STGCDB,A,INBT,SIGSG/-35.0,
2      2.0,.T.,18.0/,SIGSW,GM,NM,WM/108.0,.T.,.T.,.T./,
3      MUW1,MUW2,FW,MUW0/3162.0,3162.0,2500.0,3162.0/,KCLUT/.F./,
4      ITATTN,IATTN,MORCFAR/.F.,.F.,.F./

C      NAMELIST/INP3/MUW1,MUW2,MUW0,FW
C      NAMELIST/INP2/STKCLDB
C      NAMELIST/INP1/NPTS,WMDL,MTI,MTIMDL,THMDL,
1      STNDB,STGCDB,A,INBT,SIGSG,SIGSW,GM,NM,WM,KCLUT,NBAR,ATAY

C
C INPUTS
C
C      PRINT 801
801  FORMAT(/,10X,*INPUT INP1 - NPTS,WMDL,MTI,*,//,
1      10X,*MTIMDL,THMDL,STNDB,STGCDB,A,INBT,SIGSG,*,//,
2      10X,*SIGSW,GM,NM,WM,KCLUT,NBAR,ATAY*,/)
C      READ INP1
C      PRINT INP1

C
C INITIALIZATION
C
C      CHECK=0
C      WFLAG=0
C      DO 6 I=1,NPTS
C          SIGCSQ(I)=0.
C          SIGWSQ(I)=0.
C          SIGNSQ(I)=0.
C          WOTS(I)=1.
6      CONTINUE

C
C COMPUTE MEAN OF TARGET, NOISE AND CLUTTER
C
C      MUT=1.0
C      MUN=MUT*10.0**(-STNDB/10.0)
C      MUCL=MUT*10.0**(-STGCDB/10.0)

C
C IS MTI MODEL TO BE COMPUTED?
C
C      IF(.NOT.MTI)GO TO 50
C
C IF YES, WHICH MODEL IS TO BE USED, 1 OR 2?
C
C      IF(MTIMDL.EQ.2)GO TO 45
C
C
C MTI MODEL 1
C
C COMPUTE INITIAL COVARIANCE MATRIX [R(I,J)] FOR GROUND CLUTTER
C (NPTS+2 X NPTS+2)

```

```

C
  NPTSP2=NPTS+2
  SIGSGSQ=SIGSG**2
  TSQ=TS**2/2.0
  DO 15 I=1,NPTSP2
  DO 15 J=1,NPTSP2
    R(I,J)=EXP((-2.0*PI*SIGSG*(I-J)*T)**2)/2.0)
  15 CONTINUE
C*****
C OUTPUT [R(I,J)] FOR GROUND CLUTTER (NPTS+2 X NPTS+2)
C*****
C   PRINT 908
  908 FORMAT(//,1X,*      R MATRIX FOR GROUND CLUTTER*,/)
  DO 14 I=1,NPTSP2
  PRINT 902,(R(I,J),J=1,NPTSP2)
  14 CONTINUE
C
C INITIALIZE [B(I,J)] MATRIX FOR MODEL 1 (NPTS+2 X NPTS+2)
C
  18 NPTSP2=NPTS+2
  DO 20 I=1,NPTSP2
  DO 20 J=1,NPTSP2
    B(I,J)=0.0
  20 CONTINUE
  DO 25 I=1,NPTS
    B(I,I)=1.0
    B(I,I+1)=-2.0
    B(I,I+2)=1.0
  25 CONTINUE
    B(NPTSP2-1,NPTSP2-1)=1.0
    B(NPTSP2,NPTSP2)=1.0
C
C COMPUTE [B(I,J)] X [R(I,J)] (NPTS+2 X NPTS+2)
C
  DO 30 I=1,NPTSP2
  DO 30 J=1,NPTSP2
    PZTEMP(I,J)=0.0
    DO 30 K=1,NPTSP2
      PZTEMP(I,J)=PZTEMP(I,J)+B(I,K)*R(K,J)
    30 CONTINUE
C
C COMPUTE [BT(I,J)]
C
  DO 35 I=1,NPTSP2
  DO 35 J=1,NPTSP2
    BT(I,J)=B(J,I)
  35 CONTINUE
C
C COMPUTE [PZ(I,J)]=[B(I,J)]*[R(I,J)]*[BT(I,J)] (NPTS+2 X NPTS+2)
C
  DO 40 I=1,NPTSP2
  DO 40 J=1,NPTSP2
    PZ(I,J)=0.0
    DO 40 K=1,NPTSP2
      PZ(I,J)=PZ(I,J)+PZTEMP(I,K)*BT(K,J)
  40 CONTINUE

```

```

C
C OUTPUT [PZ(I,J)]
C
C PRINT 901
901 FORMAT(/,1X,*      PZ MATRIX   MTI MODEL 1*,/)
DO 42 I=1,NPTSP2
C PRINT 902,(PZ(I,J),J=1,NPTSP2)
902 FORMAT(/,1X,6E13.6,/,10(1X,6E13.6,))
42 CONTINUE
C
C GO TO 60
C
C NO MTI
C
C IS CONSTANT CLUTTER TO BE COMPUTED?
C
C 50 IF(.NOT.KCLUT)GO TO 57
C
C YES-INPUT SIGNAL TO CONSTANT CLUTTER IN DB AND COMPUTE CONSTANT CLUTTER
C
C PRINT 802
802 FORMAT(/,10X,*INPUT INP2 - STKCLDB*,/)
READ INP2
PRINT INP2
KCLTS=10.0**(-STKCLDB/10.0)
KCL=MUT*KCLTS
NUCLF=10.0**(-STGCDB/10.0)-KCLTS
C
C TEMPORARY STOP - A*
C
C CALL EXIT
C
C NO - COMPUTE [R(I,J)] AND [B(I,J)] MATRICES (NPTS X NPTS)
C
57 SIGSGSQ=SIGSG**2
TSQ=T**2/2.0
DO 65 I=1,NPTS
DO 65 J=1,NPTS
R(I,J)=EXP((- (2.0*PI*SIGSG*(I-J)*T)**2)/2.0)
B(I,J)=0.0
65 CONTINUE
DO 67 I=1,NPTS
DO 67 J=1,NPTS
IF(I.NE.J)GO TO 67
B(I,J)=1.0
67 CONTINUE
C*****
C OUTPUT [R(I,J)] NO CONSTANT CLUTTER
C*****
C PRINT 909
909 FORMAT(/,1X,*      NO CONSTANT CLUTTER R MATRIX*,/)
DO 66 I=1,NPTS
C PRINT 902,(R(I,J),J=1,NPTS)
66 CONTINUE

```

```

C      CHECK=1
C      GO TO 40
C
C MTI MODEL 2
C
C COMPUTE INITIAL COVARIANCE MATRIX FOR GROUND CLUTTER (NPTS X NPTS)
C
C      45 SIGSGSQ=SIGSG**2
C      TSQ=T**2/2.0
C      DO 69 I=1,NPTS
C      DO 69 J=1,NPTS
C      R(I,J)=EXP((-2.0*PI*SIGSG*(I-J)*T)**2)/2.0
C      69 CONTINUE
C*****
C OUTPUT [R(I,J)] MTI MODEL 2
C*****
C      PRINT 910
C      910 FORMAT(/,1X,*      R MATRIX MTI MODEL 2*,/)
C      DO 71 I=1,NPTS
C      PRINT 902,(R(I,J),J=1,NPTS)
C      71 CONTINUE
C
C DEFINE [B(I,J)] FOR MODEL 2 (NPTS X NPTS)
C
C      150 DO 74 I=1,NPTS
C      DO 74 J=1,NPTS
C      IF(I.EQ.J)B(I,J)=(1.0-1.0/NPTS)
C      IF(I.NE.J)B(I,J)=-1.0/NPTS
C      74 CONTINUE
C
C      CHECK=1
C
C      60 IF(CHECK.NE.1)GO TO 70
C
C COMPUTE [PZTEMP(I,J)]=[B(I,J)]*[R(I,J)] FOR MTI MODEL 2 OR NO MTI
C (NPTS X NPTS)
C
C      DO 75 I=1,NPTS
C      DO 75 J=1,NPTS
C      PZTEMP(I,J)=0.0
C      DO 75 K=1,NPTS
C      PZTEMP(I,J)=PZTEMP(I,J)+B(I,K)*R(K,J)
C      75 CONTINUE
C
C COMPUTE [BT(I,J)] FOR MTI MODEL 2 OR NO MTI (NPTS X NPTS)
C
C      DO 80 I=1,NPTS
C      DO 80 J=1,NPTS
C      BT(I,J)=B(J,I)
C      80 CONTINUE
C
C COMPUTE COVARIANCE MATRIX [PZTIL(I,J)]=[B(I,J)]*[R(I,J)]*[BT(I,J)]
C FOR MTI MODEL 2 OR NO MTI (NPTS X NPTS)

```



```

C      DO 85 I=1,NPTS
C      DO 85 J=1,NPTS
C      PZTIL(I,J)=0.0
C      DO 85 K=1,NPTS
C      PZTIL(I,J)=PZTIL(I,J)+PZTEMP(I,K)*BT(K,J)
C      85 CONTINUE
C*****
C OUTPUT [PZTIL(I,J)] FOR MTI MODEL 2 OR NO MTI
C*****
C      PRINT 903
C      903 FORMAT(//,1X,*      PZTIL MATRIX      MTI MODEL 2 OR NO MTI*,/)
C      DO 91 I=1,NPTS
C      PRINT 902,(PZTIL(I,J),J=1,NPTS)
C      91 CONTINUE
C
C      GO TO 90
C
C COVARIANCE MATRIX AFTER MTI MODEL 2 [PZTIL(I,J)] (NPTS X NPTS)
C
C      70 NPTSP2=NPTS+2
C      DO 72 I=1,NPTS
C      DO 72 J=1,NPTS
C      PZTIL(I,J)=PZ(I,J)
C      72 CONTINUE
C*****
C OUTPUT [PZTIL(I,J)] AFTER MTI
C*****
C      PRINT 913
C      913 FORMAT(//,1X,*      PZTIL MATRIX AFTER MTI*,/)
C      DO 93 I=1,NPTS
C      PRINT 902,(PZTIL(I,J),J=1,NPTS)
C      93 CONTINUE
C
C COMPUTE WEIGHTS IF DESIRED
C
C      90 IF(.NOT.IWGT)GO TO 95
C      IF(WFLAG.EQ.1)GO TO 98
C      CALL TAYLOR(WGTS,NPTS,NBAR,ATAY)
C      WFLAG=1
C*****
C OUTPUT WEIGHTS
C*****
C      98 CONTINUE
C      PRINT 921
C      921 FORMAT(//,1X,*      TAYLOR WEIGHTS*,/)
C      DO 149 I=1,NPTS
C      PRINT 922,I,WGTS(I)
C      922 FORMAT(1X,I5,E15.8)
C      149 CONTINUE
C
C DEFINE WEIGHT MATRIX [C(I,J)] (NPTS X NPTS)

```

```

C      DO 92 I=1,NPTS
      DO 92 J=1,NPTS
      C(I,J)=0.0
      IF(I.EQ.J)C(I,J)=WGTS(I)
      92 CONTINUE
C
C  DEFINE COVARIANCE MATRIX AT FFT INPUT [PW(I,J)]=[C(I,J)]*[PZTIL(I,J)]
C  * [CT(I,J)] (NPTS X NPTS)
C
      DO 94 I=1,NPTS
      DO 94 J=1,NPTS
      CT(I,J)=C(J,I)
      PZTEMP(I,J)=0.0
      DO 94 K=1,NPTS
      PZTEMP(I,J)=PZTEMP(I,J)+C(I,K)*PZTIL(K,J)
      94 CONTINUE
      DO 96 I=1,NPTS
      DO 96 J=1,NPTS
      PW(I,J)=0.0
      DO 96 K=1,NPTS
      PW(I,J)=PW(I,J)+PZTEMP(I,K)*CT(K,J)
      96 CONTINUE
C*****
C  OUTPUT [PW(I,J)]
C*****
C      PRINT 920
      920 FORMAT(//,1X,*      PW MATRIX WITH WEIGHTS  AT FFT INPUT*,/)
      DO 97 I=1,NPTS
      PRINT 902,(PW(I,J),J=1,NPTS)
      97 CONTINUE
C
      GO TO 100
C
C  NO WEIGHTS
C
      95 DO 99 I=1,NPTS
      DO 99 J=1,NPTS
      PW(I,J)=PZTIL(I,J)
      99 CONTINUE
C
      100 CONTINUE
      IF(IWGT)GO TO 102
C*****
C  OUTPUT PW
C*****
C      PRINT 904
      904 FORMAT(//,1X,*      PW MATRIX WITHOUT WEIGHTS*,/)
      DO 101 I=1,NPTS
      PRINT 902,(PW(I,J),J=1,NPTS)
      101 CONTINUE
C
C  IS GROUND CLUTTER TO BE COMPUTED?
C
      102 IF(.NOT.GM)GO TO 105

```

```

C
C YES - COMPUTE [CP(I,K)] AND [D(I,K)] FOR GROUND CLUTTER MODEL
C
      DO 110 I=1,NPTS
      DO 110 K=1,NPTS
      KUSE=K-1
      CP(I,K)=COS((2.0*PI)/NPTS*(I-1)*KUSE)
      D(I,K)=SIN((2.0*PI)/NPTS*(I-1)*KUSE)
110 CONTINUE
C*****
C OUTPUT [CP(I,J)] AND [D(I,J)] FOR GROUND CLUTTER MODEL
C*****
C      PRINT 914
      914 FORMAT(//,1X,*      CP(I,J) MATRIX FOR GROUND CLUTTER*,/)
      DO 141 I=1,NPTS
      PRINT 902,(CP(I,J),J=1,NPTS)
141 CONTINUE
C      PRINT 915
      915 FORMAT(//,1X,*      D(I,J) MATRIX FOR GROUND CLUTTER*,/)
      DO 142 I=1,NPTS
      PRINT 902,(D(I,J),J=1,NPTS)
142 CONTINUE
C
C COMPUTE DOPPLER-DEPENDENT GROUND CLUTTER VARIANCE MODULO RANDOM
C CONSTANT WHICH IS RANGE DEPENDENT (EACH CHANNEL)
C
      DO 115 K=1,NPTS
      SUM=0.0
      DO 116 I=1,NPTS
      DO 116 J=1,NPTS
      SUM=SUM+PW(I,J)*CP(I,K)*CP(J,K)+PW(I,J)*D(I,K)*D(J,K)
116 CONTINUE
      SIGCSQ(K)=SUM
115 CONTINUE
C*****
C OUTPUT SIGCSQ
C*****
      PRINT 905
      905 FORMAT(//,1X,*      SIGCSQ ARRAY*,/)
      PRINT 902,(SIGCSQ(I),I=1,NPTS)
C
C SET GROUND CLUTTER MODEL FLAG=FALSE
C
      GM=.F.
C
C IS WEATHER CLUTTER MODEL TO BE COMPUTED?
C
      IF(GM)GO TO 125
C
C NO - REINITIALIZE WEATHER PARAMETERS
C
      SIGSW=0.0
      MUW1=0.0
      MUW2=0.0
      MUW0=0.0
      FW=0.0
      DO 103 I=1,NPTS
      SIGWSQ(I)=0.0
103 CONTINUE

```

```

C
C NO WEATHER TO BE COMPUTED - GO TO NOISE MODEL
C
C      GO TO 120
C
C GROUND MODEL IS NOT TO BE CALCULATED
C
C      105 IF(.NOT.WM)GO TO 130
C
C COMPUTE [CP(I,K)] AND [D(I,K)] FOR WEATHER CLUTTER MODEL
C
C      DO 121 I=1,NPTS
C      DO 121 K=1,NPTS
C      KUSE=K-1
C      CP(I,K)=COS(2.0*PI*(I-1)*(FLOAT(KUSE)/NPTS))
C      D(I,K)=SIN(2.0*PI*(I-1)*(FLOAT(KUSE)/NPTS))
C      121 CONTINUE
C*****
C OUTPUT [CP(I,J)] AND [D(I,J)] FOR WEATHER CLUTTER MODEL
C*****
C      PRINT 916
C      916 FORMAT(//,1X,*      CP(I,J) MATRIX FOR WEATHER CLUTTER*,/)
C      DO 143 I=1,NPTS
C      PRINT 902,(CP(I,J),J=1,NPTS)
C      143 CONTINUE
C      PRINT 917
C      917 FORMAT(//,1X,*      D(I,J) MATRIX FOR WEATHER CLUTTER*,/)
C      DO 146 I=1,NPTS
C      PRINT 902,(D(I,J),J=1,NPTS)
C      146 CONTINUE
C
C COMPUTE DOPPLER DEPENDENT WEATHER CLUTTER VARIANCE MODULO A CONSTANT
C WHICH IS RANGE DEPENDENT (EACH ELEMENT)
C
C      DO 122 K=1,NPTS
C      SUM=0.0
C      DO 124 I=1,NPTS
C      DO 124 J=1,NPTS
C      SUM=SUM+PW(I,J)*CP(I,K)*CP(J,K)+PW(I,J)*D(I,K)*D(J,K)
C      124 CONTINUE
C      SIGWSQ(K)=SUM
C      122 CONTINUE
C*****
C OUTPUT SIGWSQ
C*****
C      PRINT 906
C      906 FORMAT(//,1X,*      SIGWSQ ARRAY*,/)
C      PRINT 902,(SIGWSQ(I),I=1,NPTS)
C
C SET WEATHER CLUTTER MODEL FLAG=FALSE
C
C      WM=.F.
C      GO TO 120
C
C NO WEATHER CLUTTER - COMPUTE NOISE CLUTTER
C

```

C COMPUTE [CP(I,K)] AND [D(I,K)] FOR NOISE CLUTTER

C

```
130 DO 131 I=1,NPTS
    DO 131 K=1,NPTS
        KUSE=K-1
        CP(I,K)=COS((2.0*PI)/NPTS*(I-1)*KUSE)
        D(I,K)=SIN((2.0*PI)/NPTS*(I-1)*KUSE)
```

```
131 CONTINUE
```

C\*\*\*\*\*

C OUTPUT [CP(I,J)] AND [D(I,J)] FOR NOISE CLUTTER MODEL

C\*\*\*\*\*

C PRINT 918

```
918 FORMAT(//,1X,*      CP(I,J) MATRIX FOR NOISE MODEL*,//)
```

```
DO 147 I=1,NPTS
```

C PRINT 902,(CP(I,J),J=1,NPTS)

```
147 CONTINUE
```

C PRINT 919

```
919 FORMAT(//,1X,*      D(I,J) MATRIX FOR NOISE MODEL*,//)
```

```
DO 148 I=1,NPTS
```

C PRINT 902,(D(I,J),J=1,NPTS)

```
148 CONTINUE
```

C

C COMPUTE DOPPLER DEPENDENT NOISE CLUTTER VARIANCE (EACH CHANNEL)

C

```
DO 132 K=1,NPTS
```

```
SUM=0.0
```

```
DO 133 I=1,NPTS
```

```
DO 133 J=1,NPTS
```

```
SUM=SUM+PW(I,J)*CP(I,K)*CP(J,K)+PW(I,J)*D(I,K)*D(J,K)
```

```
133 CONTINUE
```

```
SIGNSQ(K)=SUM
```

```
132 CONTINUE
```

C\*\*\*\*\*

C OUTPUT SIGNSQ

C\*\*\*\*\*

```
PRINT 907
```

```
907 FORMAT(//,1X,*      SIGNSQ ARRAY*,//)
```

```
PRINT 902,(SIGNSQ(I),I=1,NPTS)
```

```
WRITE(6,1000)MUT,NUCL,KCL,MUN,MUCLF,
```

```
+MUW1,MUW2,MUW0
```

```
1000 FORMAT(6X*MUT=*E20.8//6X*NUCL=*E20.8//6X,
```

```
+*KCL=*E20.8//6X*MUN=*E20.8//6X,
```

```
+*MUCLF=*E20.8//6X,*MUW1=*E20.8//6X,
```

```
+*MUW2=*E20.8//6X,*MUW0=*E20.8)
```

```
WRITE(2)MUT,NUCL,KCL,MUN,MUCLF,MUW1,MUW2,MUW0
```

```
WRITE(2)(SIGCSQ(K),K=1,NPTS),(SIGWSQ(K),K=1,NPTS),
```

```
+(SIGNSQ(K),K=1,NPTS)
```

```
WRITE(2)(WOTS(K),K=1,NPTS)
```

```
ENDFILE2
```

C

```
GO TO 140
```

C

C WEATHER CLUTTER MODEL

C

C WEATHER MODEL 1 OR MODEL 2?

```

C
125 IF(WMDL.EQ.2)CALL EXIT
C
C WEATHER MODEL 1
C
    PRINT 803
803 FORMAT(/,*INPUT INP3 - MUW1,MUW2,MUW0,FW*)
    READ INP3
    PRINT INP3
C
C COMPUTE [R(I,J)] MATRIX FOR WEATHER MODEL 1
C
    NPTSP2=NPTS+2
    T8Q=T**2/2.0
    SIGSW8Q=SIGSW**2
    DO 135 I=1,NPTSP2
    DO 135 J=1,NPTSP2
    R(I,J)=EXP((-2.0*PI*SIGSW*(I-J)*T)**2)/2.0)
    R(I,J)=R(I,J)*COS(2.0*PI*FW*(I-J)*T)
135 CONTINUE
C*****
C OUTPUT [R(I,J)] FOR WEATHER MODEL 1
C*****
C    PRINT 912
912 FORMAT(/,1X,*      R MATRIX  WEATHER MODEL 1*,/)
    DO 134 I=1,NPTSP2
    PRINT 902,(R(I,J),J=1,NPTSP2)
134 CONTINUE
C
C CHECK IF MTI BEING DONE
C
144 IF(MTI)GO TO 145
C
C NO MTI - DEFINE [B(I,J)]
C
    DO 138 I=1,NPTS
    DO 138 J=1,NPTS
    B(I,J)=0.0
    IF(I.EQ.J)B(I,J)=1.0
138 CONTINUE
C
    GO TO 60
C
C CHOOSE MTI MODEL 1 OR MODEL 2
C
145 IF(MTINDL.EQ.2)GO TO 150
    GO TO 18
C
C NO WEATHER CLUTTER OR WEATHER CLUTTER ALREADY DEFINED
C
C NOISE MODEL
C
120 NPTSP2=NPTS+2
    DO 123 I=1,NPTSP2
    DO 123 J=1,NPTSP2
    R(I,J)=0.0
    IF(I.EQ.J)R(I,J)=1.0
123 CONTINUE
    GO TO 144
C

```

C COMPUTE CLUTTER MAP THRESHOLD ATTENUATION AND CONVERT TO DB

C  
140 CONTINUE  
CALL EXIT  
END

C\*\*\*\*\*  
SUBROUTINE TAYLOR(GT,NGT,NBAR,A)

C  
C COMPUTATION OF THE TAYLOR ILLUMINATION FUNCTION  
C GT-ARRAY TO BE RETURNED TO MAIN PROGRAM CONTAINING TAYLOR WEIGHTS  
C NGT-DIMENSION OF GT

C  
DIMENSION GT(NGT),F(64)  
PI=3.14159265  
NBARN1=NBAR-1  
FNBAR=NBAR  
SIGSQ=FNBAR\*\*2/(A\*\*2+(FNBAR-.5)\*\*2)

C  
NGT2=NGT/2  
FNGT=NGT  
DO 100 IXP=1,NGT2  
XPR=IXP  
XPRIME=(((-FNGT/2.0))+XPR)/FNGT  
SUMFM1=0.0  
SUMFM2=0.0  
DO 50 M=1,NBARN1  
FM=M  
PROD1=1.0  
PROD2=1.0  
DO 40 N=1,NBARN1  
FN=N  
ARG1A=A\*\*2+(FN-.5)\*\*2  
ARG1=1.0-FM\*\*2/(SIGSQ\*ARG1A)  
PROD1=PROD1\*ARG1  
IF(N.EQ.M)GO TO 40  
ARG2=1.0-(FM\*\*2/FN\*\*2)  
PROD2=PROD2\*ARG2  
40 CONTINUE  
F(M)=(((1.0)\*\*(M+1))\*PROD1)/(2.0\*PROD2)  
SUMFM1=SUMFM1+F(M)\*COS(2.0\*PI\*FM\*XPRIME)  
SUMFM2=SUMFM2+F(M)  
50 CONTINUE  
GT(IXP)=(1.0+2.0\*SUMFM1)/(1.0+2.0\*SUMFM2)  
100 CONTINUE

C  
DO 200 I=1,NGT2  
IM1=I-1  
GT(NGT2+I)=GT(NGT2-IM1)  
200 CONTINUE

C  
RETURN  
END

C\*\*\*\*\*  
READY.  
BYE

R533114 LOG OFF 10.49.05.  
R533114 SRU 2.363 UNTS.

```

PROGRAM CFARP(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE2)
REAL MUT,MUCL,KCL,MUN,MUCLF,MUW1,MUW2,MUW0
INTEGER THMDL,RHOG
LOGICAL ITATTN,IATTN,IFRESP,MORCFAR,PRFA,IDEALCF
LOGICAL STIPL,MTI,IWGT,PRED
DIMENSION SIGCSQ(64),SIGWSQ(64),SIGNSQ(64),TI(64)
DIMENSION TI1(64),STI1(64),CHT(64),CFAR1(64)
DIMENSION CFAR2(64),BETA(64),PFA(64),PFASUM(64)
DIMENSION TTEST(64),E(64),PD(64),SUMPD(64)
DIMENSION STI(64),STIDB(64),SCALET(64),WGTS(64),SIG(1000)
DIMENSION TATTN(64),TATTNDB(64),FREQ(64),ATTN(64),ATTNDB(64)
DIMENSION FRESP(64),FRESPDB(64)
DATA T,PI/2.E-04,3.14159265/
DATA A/2./
DATA MUW/1000/
NAMELIST/OUT1/SIG
NAMELIST/OUT2/CFAR1,CFAR2,BETA,PFA,PFASUM,TTEST,E,PD,SUMPD,
1CHT
NAMELIST/OUT3/MUT,MUCL,KCL,MUN,MUCLF,MUW1,MUW2,MUW0
NAMELIST/OUT4/SIGCSQ,SIGWSQ,SIGNSQ,WGTS
NAMELIST/OUT5/SCALET
NAMELIST/INP1/RHOG,NRCCELLS,NCLMAP,NPTS,MTINDL
1      ITATTN,IATTN,IFRESP,MORCFAR,MTI,IWGT,STIPL,THMDL,
2      PRFA,IDEALCF,PRED
WRITE(6,10)
10 FORMAT(6X*INPUT INP1:RHOG,NRCCELLS,NCLMAP,NPTS,MTINDL,*,/
1      *ITATTN,IATTN,IFRESP,MORCFAR,MTI,IWGT,STIPL,THMDL*/
2      *PRFA,IDEALCF,PRED*)
READ INP1
REWIND2
READ(2)MUT,MUCL,KCL,MUN,MUCLF,MUW1,MUW2,MUW0
READ(2)(SIGCSQ(K),K=1,NPTS),(SIGWSQ(K),K=1,NPTS),
1(SIGNSQ(K),K=1,NPTS)
READ(2)(WGTS(K),K=1,NPTS)
140 DO 153 K=1,NPTS
IF(SIGCSQ(K).LT.0.)SIGCSQ(K)=1.E-50
IF(SIGWSQ(K).LT.0.)SIGWSQ(K)=1.E-50
IF(SIGNSQ(K).LT.0.)SIGNSQ(K)=1.E-50
TEST=MUCL*SIGCSQ(1)
IF(TEST.EQ.0.0)TEST=1.E-50
TATTN(K)=(MUCL*SIGCSQ(K))/
1TEST
TATTNDB(K)=10.0*ALOG10(TATTN(K))
IF(TATTNDB(K).LT.(-100.0))TATTNDB(K)=-100.0
153 CONTINUE
C
C CHECK IF CLUTTER MAP THRESHOLD ATTENUATION VS.FREQUENCY IS TO
C BE PLOTTED
C
C      IF(.NOT.ITATTN)GO TO 155
C
C COMPUTE FREQUENCY ARRAY

```



```
C
-   FACT=1.0/(T*NPTS)
-   DO 160 K=1,NPTS
-   FREQ(K)=(K-1)*FACT
-   160 CONTINUE
C
C PLOT CLUTTER MAP THRESHOLD ATTENUATION VS.FREQUENCY
C
-   PRINT 923
-   923 FORMAT(//,1X,*CLUTTER MAP THRESHOLD ATTENUATION VS.FREQ. PLOT*,/,
-   1 1X,*SET UP PAPER FOR 0 TO -80 DB, AND TYPE 80*)
-   READ 804,IG0
-   804 FORMAT(A2)
-   CALL PLOT(FREQ,TATTNDB,NPTS)
C
C PRINT OUT CLUTTER MAP THRESHOLD ATTENUATION
C
-   PRINT 927
-   927 FORMAT(///,1X,7X,*FREQ*,10X,*TATTN*,10X,*TATTNDB*,/)
-   DO 165 I=1,NPTS
-   PRINT 928,I,FREQ(I),TATTN(I),TATTNDB(I)
-   928 FORMAT(1X,I2,3X,F7.2,E15.8,2X,E15.8)
-   165 CONTINUE
C
C CHECK IF GROUND CLUTTER PLUS NOISE ATTENUATION VS.FREQUENCY IS TO BE PLOTTED
C
-   155 IF(.NOT.IATTN)GO TO 178
C
C COMPUTE GROUND CLUTTER ATTENUATION AND CONVERT TO DB
C
-   DO 172 K=1,NPTS
-   TEST=SIGCSQ(1)*MUCL+MUN*SIGNSQ(1)
-   IF(TEST.EQ.0.0)TEST=1.E-50
-   ATTN(K)=(MUCL*SIGCSQ(K)+MUN*SIGNSQ(K))/TEST
-   IF(ATTN(K).EQ.0.)ATTN(K)=1.E-50
-   ATTNDB(K)=10.0*ALOG10(ATTN(K))
-   IF(ATTNDB(K).LT.(-100.0))ATTNDB(K)=-100.0
-   172 CONTINUE
C
C COMPUTE FREQUENCY ARRAY
C
-   FACT=1.0/(T*NPTS)
-   DO 174 K=1,NPTS
-   FREQ(K)=(K-1)*FACT
-   174 CONTINUE
C
C PLOT GROUND CLUTTER PLUS NOISE ATTENUATION VS.FREQUENCY
C
-   PRINT 929
-   929 FORMAT(//,1X,*GROUND CLUTTER ATTENUATION VS.FREQ.PLOT*,/,
-   1 1X,*SET UP PAPER FOR 0 TO -80 DB, AND TYPE 80*)
-   READ 804,IG0
-   CALL PLOT(FREQ,ATTNDB,NPTS)
C
```

C PRINT OUT GROUND CLUTTER ATTENUATION

C

```

      PRINT 930
930  FORMAT(///,1X,7X,*FREQ*,10X,* ATTN*,10X,* ATTNDB*,/)
      DO 177 I=1,NPTS
      PRINT 928,I,FREQ(I),ATTN(I),ATTNDB(I)
177  CONTINUE
178  IF(.NOT.IFRESP)GO TO 170
      DO 171 K=1,NPTS
      TEST=SIGNSQ(1)
      IF(TEST.EQ.0.0)TEST=1.E-50
      FRESP(K)=SIGNSQ(K)/TEST
      IF(FRESP(K).EQ.0.)FRESP(K)=1.E-50
      FRESPDB(K)=10.*ALOG10(FRESP(K))
      IF(FRESPDB(K).LT.-100.)FRESPDB(K)=-100.
171  CONTINUE
      FACT=1./(T*NPTS)
      DO 176 K=1,NPTS
      FREQ(K)=(K-1)*FACT
176  CONTINUE

```

C

C

C

PLOT OF FILTER RESPONSE

```

      PRINT 183
183  FORMAT(//,1X,*FREQUENCY RESPONSE VS TOT VEL*)
      READ 804,IG0
      CALL PLOT(FREQ,FRESPDB,NPTS)

```

C

C

C

CHECK TO SEE IF CFAR TO BE CONTINUED

```

      IF(.NOT.NORCFAR)STOP11

```

C

C

C

COMPUTE SIGNAL-TO-INTERFERENCE VS.TARGET DOPPLER

C

```

170  DO 182 K=1,NPTS
      TI(K)=MUCL*SIGCSQ(K)+MUW0*SIGWSQ(K)+MUN*SIGNSQ(K)
182  CONTINUE

```

C\*\*\*\*\*

C OUTPUT TI(K)

C\*\*\*\*\*

C PRINT 931

```

931  FORMAT(//,1X,*      TI ARRAY*,/)

```

C

```

      PRINT 902,(TI(K),K=1,NPTS)

```

```

902  FORMAT(/,1X,6E13.6,/,10(1X,6E13.6,/)

```

C

```

      CALL TSCALE(WGTS,MTI,MTINDL,NPTS,T,SCALET,INGT)

```

C

```

      WRITE OUT5

```

C

```

      DO 185 K=1,NPTS

```

```

      TEST=TI(K)

```

```

      IF(TEST.EQ.0.)TEST=1.E-50

```

```

      STI(K)=(MUT*SCALET(K))/TEST

```

```

      IF(STI(K).EQ.0.)STI(K)=1.E-50

```

```

      STIDB(K)=10.0*ALOG10(STI(K))

```

```

      IF(STIDB(K).GT.100.0)STIDB(K)=100.0

```

```

      IF(STIDB(K).LT.(-100.0))STIDB(K)=-100.0

```

```

185  CONTINUE
    IF(.NOT.STIPL)GO TO 4010
    FACT=1./(T*NPTS)
    DO 4020 K=1,NPTS
    FREQ(K)=(K-1)*FACT
4020  CONTINUE
    PRINT 1000
1000  FORMAT(/,1X,$SIGNAL-TO-INTERFERENCE PLOT*)
    READ 804,IG0
    CALL PLOT(FREQ,STIDB,NPTS)
4010  CONTINUE
C
C  TEMPORARY STOP
C
180  CONTINUE
    IF(THMDL.EQ.1)GO TO 5000
    STOP13
5000  IF(RHOG.EQ.1)GO TO 5010
    STOP15
C  CLUTTER MAP THRESHOLD CONSTANT
5010  XKM=10.**(-6./NCLMAP)
    XKM=NCLMAP/XKM-NCLMAP
    WRITE(6,5020)XKM
5020  FORMAT(/,6X$CLUTTERMAP FALSE ALARM CONSTANT=$E20.8)
C  CFAR CONSTANT DETERMINATION
    XNRD2=NRCELLS/2.
    XKCFAR=10.**(-6./XNRD2)
    XKCFAR=XNRD2*(XKCFAR-1.)
    WRITE(6,5050)XKCFAR
5050  FORMAT(,6X$CFAR FALSE ALARM CONSTANT=$E20.8)
C  AVG CLUTTERMAP THRESHOLDS
    CALL WBULL1(MUCL,A,SIG)
C
    WRITE OUT1
    DO 5061 K=1,NPTS
    PFASUM(K)=0.
    SUMPD(K)=0.
5061  CONTINUE
    DO 5030 NW=1,NW
    DO 5040 K=1,NPTS
    TI1(K)=SIG(NW)*SIGCSQ(K)+MUW0*SIGWSQ(K)+MUN*SIGNSQ(K)
    TEST=TI1(K)
    IF(TEST.EQ.0.)TEST=1.E-50
    STI1(K)=MUT$SCALET(K)/TEST
    CMT(K)=TI1(1)*TATTN(K)
5040  CONTINUE
C  AVG CFAR THRESHOLDS
    DO 5060 K=1,NPTS
    CFAR1(K)=(MUW1-MUW0)*SIGWSQ(K)+TI1(K)
    CFAR1(K)=CFAR1(K)*XKCFAR
    CFAR2(K)=TI1(K)+(MUW2-MUW0)*SIGWSQ(K)
    CFAR2(K)=CFAR2(K)*XKCFAR
5060  CONTINUE
    DO 5080 K=1,NPTS
    TEST=TI1(K)
    IF(TEST.EQ.0.)TEST=1.E-50
    BETA(K)=TATTN(K)*TI1(1)/TEST
    PFA(K)=(NCLMAP/(NCLMAP+XKM*BETA(K)))*NCLMAP
    PFASUM(K)=PFA(K)+PFASUM(K)

```

```

5080 CONTINUE
6001 DO 6000 K=1,NPTS
      TTEST(K)=AMAX1(CMT(K),CFAR1(K),CFAR2(K))
      IF(K.EQ.1)GO TO 6010
      IF(TTEST(K).EQ.CMT(K))GO TO 6010
      IF(TTEST(K).EQ.CFAR1(K))GO TO 6030
      NTHOLD=3
      GO TO 6100
6010 NTHOLD=1
      GO TO 6100
6030 NTHOLD=2
6100 GO TO(6110,6120,6130),NTHOLD
6110 XKP=XKM
      NP=NCLMAP
      E(K)=BETA(K)
      GO TO 6140
6120 NP=XNRD2
      XKP=XKCFAR
      TEST=TI1(K)
      IF(TEST.EQ.0.)TEST=1.E-50
      E(K)=CFAR1(K)/(XKCFAR*TEST)
      GO TO 6140
6130 XKP=XKCFAR
      NP=XNRD2
      TEST=TI1(K)
      IF(TEST.EQ.0.)TEST=1.E-50
      E(K)=CFAR2(K)/(XKP*TEST)
6140 PD(K)=1./((1.+(XKP*E(K))/(NP*(1.+8*TI1(K))))**NP)
      SUMPD(K)=SUMPD(K)+PD(K)
6000 CONTINUE
C   IF(NW.LE.2)WRITE OUT2
5030 CONTINUE
5090 DO 6160 K=1,NPTS
      FREQ(K)=(K-1.)*1./(T*NPTS)
6160 CONTINUE
      DO 6150 K=1,NPTS
          PFASUM(K)=PFASUM(K)/NWX
          SUMPD(K)=SUMPD(K)/NWX
6150 CONTINUE
      IF(.NOT.PRFA)GO TO 6200
C   PROBABILITY OF FALSE ALARM VS FREQUENCY
      WRITE(6,6169)
6169 FORMAT(6X*PROB. OF FALSE ALARM VS FREQUENCY*)
      READ 804,IG0
      CALL PLOT(FREQ,PFASUM,NPTS)
      WRITE(6,6170)
6170 FORMAT(//6X*FREQUENCY*10X*PROB OF FALSE ALARM*//)
      WRITE(6,6310)(FREQ(K),PFASUM(K),K=1,NPTS)
C   PROBABILITY OF DETECTION
6200 CONTINUE
      IF(.NOT.PRED)GO TO 7001
      WRITE(6,6210)
6210 FORMAT(6X*GRAPH OF DETECTION PROBABILITY*)
      READ 804,IG0
      CALL PLOT(FREQ,SUMPD,NPTS)
      WRITE(6,6300)
6300 FORMAT(//6X*FREQUENCY*10X*DETECTION PROBABILITY*//)
      WRITE(6,6310)(FREQ(K),SUMPD(K),K=1,NPTS)

```

```

6310 FORMAT(1X,E20.8,4X,E20.8)
7001 IF(.NOT.IDEALCF)GO TO 7000
    YB=13.81551056
    DO 7090 K=1,NPTS
        SUMPD(K)=0.
7090 CONTINUE
    DO 8000 NW=1,NWW
        DO 8010 K=1,NPTS
            TI1(K)=SIG(NW)*SIGCSQ(K)+MUWO*SIGWSQ(K)+MUN*SIGNSQ(K)
            TEST=TI1(K)
            IF(TEST.EQ.0.)TEST=1.E-50
            STI(K)=MUT*SCALET(K)/TEST
            PD(K)=EXP(-YB/(1.+STI(K)))
            SUMPD(K)=SUMPD(K)+PD(K)
8010 CONTINUE
8000 CONTINUE
        DO 8020 K=1,NPTS
            SUMPD(K)=SUMPD(K)/NWW
8020 CONTINUE
        WRITE(6,8030)
8030 FORMAT(6X*PROBABILITY OF DETECTION:IDEAL CFAR*)
        READ 804,IG0
        CALL PLOT(FREQ,SUMPD,NPTS)
        WRITE(6,6300)
        WRITE(6,6310)(FREQ(K),SUMPD(K),K=1,NPTS)
7000 CONTINUE
        CALL EXIT
        END
C*****
C*****
      SUBROUTINE TSCALE(WGTS,MTI,MTINDL,NPTS,T,SCALET,IWGT)
C
C
C      LOGICAL IWGT,MTI
C
C      COMPLEX X,XTEMP,SUM,AVGX,XBAR
C
C      DIMENSION WGTS(64),SCALET(64),X(66),XTEMP(64),XBAR(64),FREQ(64)
C
C
C      PI=3.14159265
C
C      COMPUTE FREQUENCY ARRAY
C
C      FACT=1.0/(T*NPTS)
C      DO 10 K=1,NPTS
C          FREQ(K)=(K-1)*FACT
C      10 CONTINUE
C
C      COMPUTE X(L),L=1,NPTSP2 FOR EACH K
C
C      NPTSP2=NPTS+2
C
C      BEGINNING OF K LOOP
C

```

```

      DO 15 K=1,NPTS
C
      DO 20 L=1,NPTSP2
      ARG=2.0*PI*FREQ(K)*(L-1)*T
      X(L)=CMPLX(COS(ARG),SIN(ARG))
      20 CONTINUE
C*****
C OUTPUT X(L) FOR EACH K
C*****
C      PRINT 900,K
      900 FORMAT(//,*      X(L) FOR K=*,I2,/)
C      PRINT 902,(X(L),L=1,NPTSP2)
      902 FORMAT(/,1X,2E13.6,2X,2E13.6,/,31(1X,2E13.6,2X,2E13.6,/))
C
C CHECK IF MTI TO BE DONE
C
      IF(.NOT.MTI)GO TO 30
C
C IF MTI TO BE DONE, MODEL #1 OR MODEL #2 ?
C
      IF(MTINDL.EQ.2)GO TO 25
C
C MTI MODEL #1
C
      DO 22 L=1,NPTS
      XTEMP(L)=X(L)-2.0*X(L+1)+X(L+2)
      22 CONTINUE
      DO 23 L=1,NPTS
      X(L)=XTEMP(L)
      23 CONTINUE
C*****
C OUTPUT X(L) FOR EACH K AFTER MTI MODEL #1
C*****
C      PRINT 903,K
      903 FORMAT(//,1X,*      X(L) AFTER MTI MODEL #1 FOR K=*,I2,/)
C      PRINT 902,(X(L),L=1,NPTS)
C
C      GO TO 30
C
C MTI MODEL #2
C
      25 SUM=0.0
      DO 35 L=1,NPTS
      SUM=SUM+X(L)
      35 CONTINUE
      AVGX=SUM/NPTS
      DO 37 L=1,NPTS
      X(L)=X(L)-AVGX
      37 CONTINUE
C*****
C OUTPUT X(L) FOR EACH K AFTER MTI MODEL #2
C*****
C      PRINT 904,K
      904 FORMAT(//,1X,*      X(L) AFTER MTI MODEL #2 FOR K=*,I2,/)
C      PRINT 902,(X(L),L=1,NPTS)
C

```

```

C APPLY WEIGHTS IF REQUIRED
C
  30 IF(.NOT.IWGT)GO TO 40
    DO 39 L=1,NPTS
      X(L)=X(L)*WGT(L)
    39 CONTINUE
C*****
C OUTPUT X(L) FOR EACH K AFTER WEIGHTING
C*****
C   PRINT 905,K
  905 FORMAT(/,1X,*      X(L) AFTER WEIGHTING FOR K=*,I2,/)
C   PRINT 902,(X(L),L=1,NPTS)
C
C COMPUTE XBAR(K)
C
  40 SUM=0.0
    DO 45 L=1,NPTS
      ARG=(2.0*PI)/NPTS*((L-1)*(K-1))
      SUM=SUM+X(L)*CMPLX(COS(ARG),-SIN(ARG))
    45 CONTINUE
    XBAR(K)=SUM
C*****
C OUTPUT XBAR(K)
C*****
C   PRINT 906,K,XBAR(K)
  906 FORMAT(/,1X,*K=*,I2,2X,*XBAR(K)=*,2E13.6,/)
C
C DEFINE OUTGOING SCALET ARRAY
C
  SCALET(K)=(REAL(XBAR(K)))*2+(AIMAG(XBAR(K)))*2
C*****
C OUTPUT SCALET(K)
C*****
C   PRINT 907,K,SCALET(K)
  907 FORMAT(/,1X,*K=*,I2,2X,*SCALET(K)=*,E13.6,/)
C
C END OF K LOOP
C
  15 CONTINUE
C
  RETURN
END
SUBROUTINE PLOT(X,Y,N)
  LOGICAL IPROB
  DIMENSION X(64),Y(64)
  DIMENSION TEMP(64)
  NAMELIST/INP1/YMIN,YMAX
  NAMELIST/INP2/IPROB,XMIN,XMAX
  NAMELIST/INP3/XMIN,XMAX,YMIN,YMAX
  DO 1 I=1,N
    TEMP(I)=Y(I)
  1 CONTINUE
  XMIN=X(1)
  XMAX=X(1)
  YMIN=TEMP(1)
  YMAX=TEMP(1)
  DO 10 I=1,N

```

```
DIFX1=X(I)-XMIN
DIFX2=X(I)-XMAX
DIFY1=TEMP(I)-YMIN
DIFY2=TEMP(I)-YMAX
IF(DIFX1.LT.0.)XMIN=X(I)
IF(DIFX2.GT.0.)XMAX=X(I)
IF(DIFY1.LT.0.)YMIN=TEMP(I)
IF(DIFY2.GT.0.)YMAX=TEMP(I)
10 CONTINUE
WRITE INP3
WRITE(6,11)
11 FORMAT(6X*INPUT INP2:IPROB,XMIN,XMAX*)
READ INP2
IF(.NOT.IPROB)GO TO 12
CALL PSCALE(TEMP,N,YMIN,YMAX)
GO TO 13
PRINT INP1
12 WRITE(6,20)
20 FORMAT(6X*INP NAMELIST INP1:YMIN,YMAX*)
READ INP1
13 PRINT 924
924 FORMAT(1X,$PLTL*)
NN=N/2.+1.
DO 30 I=1,NN
IX=(X(I)-XMIN)*9999.0/(XMAX-XMIN)
IY=(TEMP(I)-YMIN)*9999.0/(YMAX-YMIN)
PRINT 925,IX,IY
925 FORMAT(2(1X,I4))
30 CONTINUE
PRINT 926
926 FORMAT(1X,$PLTT*)
RETURN
END
SUBROUTINE WBULL1(MUCL,A,SIG)
REAL MUCL
DIMENSION SIG(1000)
DIMENSION R(1000)
DO 10 I=1,1000
R(I)=RANF(0)
SIG(I)=(ALOG(1./(1.-R(I))))**A
SIG(I)=MUCL*SIG(I)/2.
10 CONTINUE
SUM=0.
DO 20 I=1,1000
SUM=SUM+SIG(I)
20 CONTINUE
SUM=SUM/1000.
DO 30 I=1,1000
SIG(I)=MUCL*SIG(I)/SUM
30 CONTINUE
WRITE(6,50)SUM
```

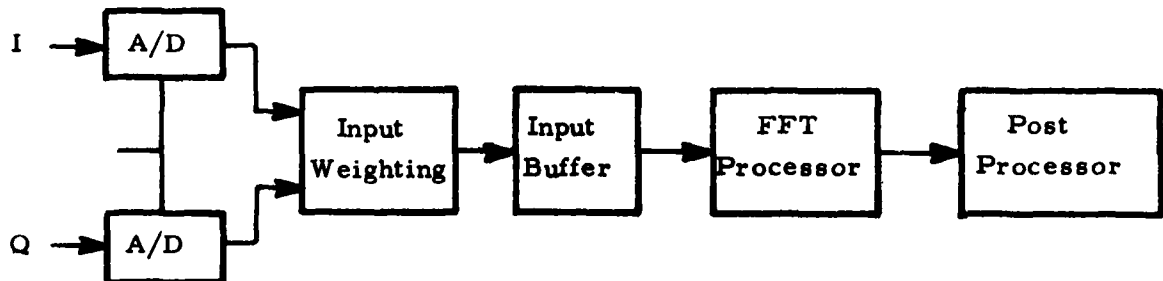


```
50 FORMAT(6X*MEAN OF UNSCALED CLUTTER CROSS SECTION SEQ IS#E20.8)
RETURN
END
SUBROUTINE PSCALE(Y,NPTS,YMIN,YMAX)
DIMENSION Y(64)
FI(P)=(A0+A1*P+A2*(P**2))/(1.+B1*P+B2*(P**2)+B3*(P**3))
A0=2.515517
A1=.802853
A2=.010328
B1=1.432788
B2=.189269
B3=.001308
AS=.05
PS=SQRT(ALOG(1./(AS**2)))
YMIN=FI(PS)
YMIN=PS-YMIN
YMIN=-YMIN
AL=.0001
PL=SQRT(ALOG(1./(AL**2)))
YMAX=FI(PL)
YMAX=PL-YMAX
DO 10 I=1,NPTS
IF(Y(I).LE.0.5)GO TO 20
AA=1.-1.E-10
IF(Y(I).EQ.1.)Y(I)=AA
P=SQRT(ALOG(1./((1.-Y(I))**2)))
Y(I)=FI(P)
Y(I)=P-Y(I)
GO TO 10
20 IF(Y(I).EQ.0.)Y(I)=1.E-10
P=SQRT(ALOG(1./(Y(I)**2)))
Y(I)=FI(P)
Y(I)=P-Y(I)
Y(I)=-Y(I)
10 CONTINUE
RETURN
END
READY.
```

APPENDIX B

USE OF EXISTING FAST FOURIER TRANSFORM DIGITAL  
PROCESSOR (FFTDP) FOR EAR

A block diagram representation of the FFTDP is depicted below. As shown it consists of a pair of 9-bit A/D converters, a time weighting circuit, an input storage buffer, and a complex adder/subtractor/multiplier arranged in an FFT butterfly configuration which outputs to a post processor.



Parameters of the FFTDP are:

1. 9-bit A/D with a 2.4  $\mu$ sec (integrate and dump) sample rate.
2. Input buffer of 512 words of 48 bits.
3.  $(\text{Cosine})^2$  time weighting circuits.
4. 64 point FFT (complex adder/complex subtractor/complex multiplier butterfly configuration).
5. 30 range gates plus 1 test gate.
6. Fixed 4.608 ms time to process a 64 point batch.

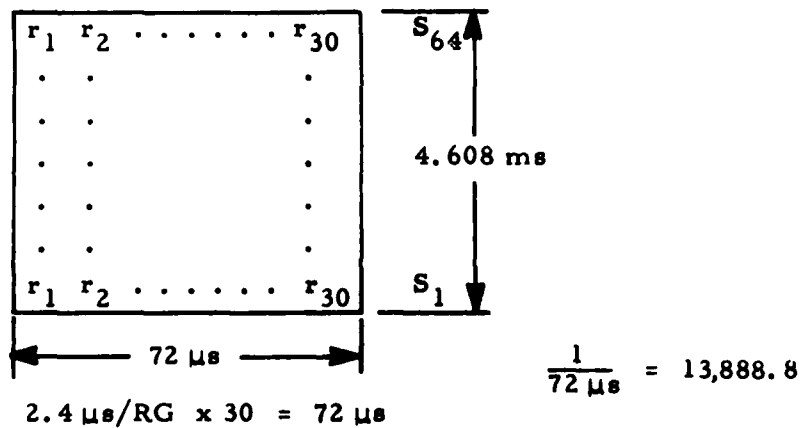
The organization of the input buffer is depicted below; it is made up of 512 words of 48 bits. Each 48-bit word is made up of 4 data words. Hence, the FFT arithmetic unit carries only 12 bits with a scaling capability when the arithmetic overflows the 12 bits.

Address	Odd	Even	Odd	Even
0	$S_1 R_1$	$S_1 R_2$	$S_{33} R_1$	$S_{33} R_2$
1	$S_1 R_3$	$S_1 R_4$	$S_{33} R_3$	$S_{33} R_4$
2	$S_1 R_5$		$S_{33} R_5$	
•				
•				
13	$S_1 R_{29}$	$S_1 R_{30}$	$S_{33} R_{29}$	$S_{33} R_{30}$
BITE { 14	Spare	Spare	Spare	Spare
15	Spare	Spare	Spare	Spare
•				
•				
509	$S_{32} R_{29}$	$S_{32} R_{30}$	$S_{64} R_{29}$	$S_{64} R_{30}$
510	Spare	Spare	Spare	Spare
511	Spare	Spare	Spare	Spare

S = Sample  
R = Range Gate

### Memory Organization

The relationships within the buffer/sample/FFT process are shown pictorially below:



process time = 4.608 ms

$$\text{time of 1 Butterfly} = \frac{4608}{\frac{64}{2} \log_2 64} = \frac{4608}{5760} = 800 \text{ ns}$$

To use this processor with the EAR, a new synchronizer must be designed as well as another input buffer suitable for reformatting the input data into blocks of 30 range gates. The 64 point input data is collected by the EAR in  $64 \times 200 = 12,800 \mu\text{sec}$  at its nominal 5 kHz PRF. The FFTDP processes a batch of 30 range gates worth of data in  $4608 \mu\text{sec}$ . Therefore  $12,800/4,608 = 2.777$  batches can be processed. This must be rounded downward to the nearest integer, so that  $2 \times 30 = 60$  range gates can be handled by the processor at the 5 kHz PRF. It is probably sensible to lower the maximum PRF to 4629 Hz. The data collection time then goes up to  $64 \times 216 = 13,826 \mu\text{sec}$ ., which will then handle exactly three batches, for a total of 90 range gates.

With such a small number of range gates, the kind of CFAR processing to be used is open to question. A sliding-window cell-averaging CFAR as discussed in the text may not make too much sense, since end effects will tend to dominate.

# DISTRIBUTION

No. of Copies

Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12
Commander U.S. Army Materiel Development and Readiness Command ATTN: DRCCG DRCRD 5001 Eisenhower Avenue Alexandria, Virginia 22304	1 1
Commander Ballistic Missile Defense Systems Command ATTN: BMDSC-HR, Mr. Stevenson P.O. Box 1500 Huntsville, Alabama 35807	1
Director Ballistic Missile Defense Advanced Technology Center ATTN: ATC-R, Mr. Carlson P.O. Box 1500 Huntsville, Alabama 35807	2
Commander US Army Electronics Command ATTN: DRSEL, Mr. Fishbien DRCPM-MALR Fort Monmouth, New Jersey	1 1
DRCPM-MDE, Mr. Wood	1
DRCPM-HAE, Mr. Ams	1
DRCPM-CFE, Mr. David	1
DRCPM-SHO, Mr. Bishop	1
DRSMI-R, Dr. McDaniel	1
-R, Mr. Fagan	1
-R, Dr. Kobler	1
-RE, Mr. Lindberg	1
-RE, Mr. Pittman	1
-REO, Mr. Currie	1
-RER, Mr. Cash	1
-REG, Dr. Burlage	50
-RE, Record Set	1
-RG, Mr. Huff	1
-RD, Dr. McCorkle	1
-RBL	5
-RP, Dr. Jackson	1